dr hab. inż. Paweł T. Wojciechowski
Instytut Informatyki
Politechnika Poznańska
ul. Piotrowo 2, 60-965 Poznań
tel.: 61 665 3021
e-mail: Pawel.T.Wojciechowski@put.edu.pl

Poznań, 25 października 2014 r.

RECENZJA ROZPRAWY DOKTORSKIEJ

mgra Marka Materzoka

zatytułowanej

Control Abstraction for Layered Continuations. Semantics, Types and Implementation

Niniejsza recenzja została przygotowana na zlecenie Dziekana Wydziału Matematyki i Informatyki Uniwersytetu Wrocławskiego Pana Prof. dr hab. Piotra Bilera (pismo z dnia 23 czerwca 2014 r.). Recenzja dotyczy rozprawy doktorskiej mgra Marka Materzoka, zatytułowanej: *Control Abstraction for Layered Continuations. Semantics, Types and Implementation* (tytuł w języku polskim: *Abstrakcja sterowania dla kontynuacji wielowarstwowych. Semantyka, typy i implementacja*).

Recenzowana rozprawa została napisana w języku angielskim i składa się z dziewięciu rozdziałów oraz pięciu apendyksów z dowodami twierdzeń. Zawiera także jednostronicowe streszczenia w języku angielskim i polskim, spis treści, listę rysunków, oraz bibliografię liczącą 92 pozycje. Ogółem praca liczy 121 stron. Struktura rozdziałów jest przejrzysta. Na uwagę zasługuje wysoka dbałość o stronę redakcyjną pracy oraz poprawność językową. Praca jest wybitnie teoretyczna, wobec czego dominują zapisy formalne, a opis w języku naturalnym jest mocno skondensowany. Z drugiej strony, w większości rozdziałów nie odczuwa się niedostatku objaśnień do opisów formalnych, co łącznie wskazuje na bardzo dobre opanowanie przez autora warsztatu naukowego w zakresie informatyki teoretycznej.

Praca dotyczy formalnej semantyki języków programowania, a konkretnie tej jej części, która opisuje abstrakcje sterowania dla kontynuacji wielowarstwowych (*ang.* layered continuations albo delimited continuations). W szczególności, Autor rozwija i bada język operatorów sterowania $\mathsf{shift}_0/\mathsf{reset}_0$, które są wariantem operatorów sterowania shift/reset zaproponowanych przez Danvy i Filińskiego. W istocie, Autor prezentuje w rozprawie dwa języki: λ_{S_0}, który jest oparty na operatorach sterowania shift_0 i reset_0 oraz jego wariant, zwany $\lambda_\$$, który zawiera operatory shift_0 oraz \$, gdzie ten ostatni operator jest uogólnieniem operatora reset_0. \$ jest wariantem podobnego operatora, który opisali Kiselyov i Shan. Oba języki bazują na klasycznym rachunku lambda z typami i wołaniem przez wartość (*ang.* typed call-by-value λ-calculus), rozszerzając go o wspomniane wyżej operatory. W dalszej części recenzji, pisząc "język" bez podania jego nazwy, odnoszę się do obu języków łącznie.

Oprócz zdefiniowania semantyki operacyjnej i maszyny abstrakcyjnej dla opracowanego języka, zaproponowano także system typów (z podtypowaniem) oraz translację CPS (*ang.*

continuation-passing style). W kolejnym kroku Autor wszechstronnie przebadał opracowany przez siebie język w sposób formalny. Przykładowo, praca zawiera porównanie opracowanego systemu typów i translacji CPS z istniejącymi odpowiednikami, które zostały zaproponowane dla operatorów shift/reset. Ponadto, Autor przebadał język $\lambda_{\$}$ w kontekście hierarchii operatorów sterowania dla kontynuacji wielowarstwowych (zaproponowanej przez Biernacka, Biernacki, i Lenglet) oraz zaprezentował odpowiednią aksjomatyzację równaniową (*ang.* equational axiomatization). Na koniec Autor opisał własną implementację operatorów $\mathsf{shift}_0/\mathsf{reset}_0$ w asemblerze x86-64, mającą formę biblioteki z interfejsem dla języka C/C++. Dla ilustracji możliwych zastosowań biblioteki pokazano dwa przykładowe programy korzystające z operatorów $\mathsf{shift}_0/\mathsf{reset}_0$.

Na stronie drugiej sformułowano główny cel rozprawy, tj. opracowanie teorii operatorów sterowania $\mathsf{shift}_0/\mathsf{reset}_0$ uogólniającej wcześniej znane rezultaty otrzymane dla shift/reset oraz hierarchii CPS. Cel ten został w pełni osiągnięty, a ponadto wypracowane wyniki teoretyczne mają też istotne znaczenie praktyczne. Najważniejszy i oryginalny dorobek Autora, stanowiący jednocześnie główną kontrybucję rozprawy, jest następujący. Po pierwsze, zaproponowano nowy operator \$ oraz udowodniono, że pary operatorów $\mathsf{shift}_0/\$$ oraz $\mathsf{shift}_0/\mathsf{reset}_0$ mają tę samą ekspresywność, co pozwala używać tę pierwszą parę operatorów aby wykazać własności tej drugiej pary. W szczególności udowodniono, że operatory hierarchii CPS mogą być wyrażone używając $\mathsf{shift}_0/\mathsf{reset}_0$. Drugą kontrybucją jest zdefiniowanie systemu typów dla $\mathsf{shift}_0/\mathsf{reset}_0$ wraz z podtypowaniem. System ten pozwala charakteryzować możliwe efekty sterowania dla wyrażeń w zaproponowanym języku. Od strony formalnej, zdefniowany system typów generalizuje systemy typów zaproponowane wcześniej dla shift/reset oraz hierarchii CPS. Po trzecie, zdefiniowano translację CPS ukierunkowaną na typy, która ma własność selektywności, tj. nie transformuje wyrażeń które nie mają efektów sterowania, co pozwala na efektywną implementację operatorów $\mathsf{shift}_0/\mathsf{reset}_0$. Po czwarte, Autor zdefiniował też aksjomaty dla $\mathsf{shift}_0/\mathsf{reset}_0$ oraz udowodnił ich poprawność i kompletność w odniesieniu do typowanych i nietypowanych translacji CPS. Typowana wersja tej teorii generalizuje aksjomaty dla shift/reset, które wcześniej zaproponowali Kameyama i Hasegawa. Po piąte, w celu pokazania praktycznego zastosowania dla swoich operatorów sterowania, Autor opracował testową implementację dla języka C/C++.

Stwierdzam też, że recenzowana praca ma charakter naukowy i może być przedmiotem rozprawy doktorskiej z dziedziny informatyki. Wyniki uzyskane przez Autora rozprawy w istotny sposób uzupełniają istniejący stan wiedzy. Należy więc uznać podjęcie wspomnianego problemu badawczego za w pełni uzasadnione. Jakkolwiek trzeba też dodać w tym miejscu, że Autor nie odkrywał w swojej rozprawie zupełnie nowych obszarów badawczych, gdyż w dużej mierze mógł posiłkować się dotychczasowymi wynikami, które zostały opracowane dla podobnych operatorów shift/reset oraz wcześniejszymi pracami promotora pomocniczego na temat kontynuacji ograniczonych.

Autor zastosował w pracy poprawną terminologię oraz odniósł się do aktualnego stanu wiedzy w zakresie badanej tematyki przez analizę istniejących rozwiązań, posługując się przy tym licznymi odwołaniami do literatury. Na uwagę zasługuje dbałość Autora o precyzyjne

definiowanie pojęć oraz ilustrowanie omawianych zagadnień przykładami.

Wyniki zaprezentowane w rozprawie zostały częściowo opublikowane w materiałach czterech konferencji lub warsztatów (w tym bardzo dobrej konferencji ICFP), a także w jednym czasopiśmie Springera *Higher Order Symbolic Computation.* Na marginesie należy dodać, że jakkolwiek jest to czasopismo dobrze znane i uznane w środowisku osób zajmujących się semantyką języków programowania, to jednak nie występuje ono w ministerialnym wykazie czasopism naukowych z 2013 r., posiadających *impact factor* w bazie Journal Citation Reports (JCR).

Pomimo ogólnej, wysokiej oceny pracy, Autorowi nie udało się uniknąć uchybień oraz nieścisłości, np.:

1. Strona 2, paragraf 4: "puts a new empty context on the context stack end resumes the execution of e." Powinno być: puts a new empty context on the context stack and resumes the execution of e.
2. Strona 4, paragraf 3: "Another interesting point is that the defining CPS translation for shift/reset can be derived from the type-directed translation." Bez pierwszego "the" lub "the defined" zamiast "the defining".
3. Strona 9, sekcja 2.1.3, pierwsze zdanie: "The abstract machine for $\lambda_{\mathcal{S}}$ is presented in Figure 2.2." Powinno być: The abstract machine for $\lambda_{\mathcal{S}}$ is presented in Figure 2.1.
4. strona 11, sekcja 2.2.3, drugie zdanie: "It differs from the machine for $\lambda_{\mathcal{S}}$ presented before in Figure 2.1 in the transition for shift_0: where the transition for shift left an empty context on the top of the context stack in place of the captured one, the transition for shift_0 exposes the evaluation context below". Drobna uwaga: Dodałbym do tego długiego objaśnienia, że technicznie oznacza to jedynie tyle, że w definicji nowej maszyny abstrakcyjnej zamieniamy w jednej regule $\mathcal{S}$ na $\mathcal{S}_0$, a pozostałe reguły pozostają bez zmian.
5. Strona 12, sekcja 2.3, drugie zdanie: "In $\lambda_{\mathcal{S}_0}$ the reset_0 control operator is replaced with its generalized version, the binary operator \$." Powinno być: In $\lambda_{\$}$ the reset_0 control operator is replaced with its generalized version, the binary operator \$.
6. Strona 14, sekcja 2.3.2, ostatnie zdanie: "The $\lambda_{\$}$ language has the unique decomposition property, which is stated the same way as for $\lambda_{\mathcal{S}_0}$ (Lemma 2.2)." Powinno być: The $\lambda_{\$}$ language has the unique decomposition property, which is stated the same way as for $\lambda_{\mathcal{S}_0}$ (Property 2.2).
7. Strona 14, przedostatnia linia: "The second macro-expresses the first in a more complicated way:". Lepiej: The second operator macro-expresses the first in a more complicated way. W przeciwnym razie można błędnie odczytać to zdanie jako: The second macro expresses (..). Poza tym zamiast the first/the second bardziej tu pasuje the former/the latter.
8. Strona 21, sekcja 3.3.1: "The type system with subtyping for $\lambda_{\mathcal{S}_0}$, called $\lambda^{\leq}_{\mathcal{S}_0}$, is shown in Figure 3.3.". W istocie Figure 3.3 prezentuje także dwie dodatkowe reguły typowania dla języka $\lambda_{\$}$, więc jest to system typów dla obu języków (co jest zgodnie z opisem

poniżej Figure 3.3).

9. Strona 23, ostatni paragraf: "The type system has two distinct rules for function application: one for pure expressions (i.e., those without control effects, and one for impure expressions." Brakuje podania nazw tych reguł, odnosząc się do definicji systemu typów podanej na Figure 3.3. W domyśle, pierwsza reguła to PAPP a druga to APP.
10. Dalej Autor pisze: "The rule for pure application seems not strictly necessary, but its removal reduces the expressiveness of the type system – for example, the following term

$$(\lambda f.\lambda y.(\lambda z.\langle(\lambda v.\lambda w.y)(f\ y)\rangle)\ \langle f\ y\rangle)\ (\lambda x.(\lambda x.x)\ x)$$

can no longer be typed without this rule." Następnie Autor wyjaśnia, że bez wspomnianej dodatkowej reguły, aplikacja funkcji wewnątrz podwyrażenia $\lambda x.(\lambda x.x)\ x$ byłaby typowana jako "impur" (domyślnie: używając reguły APP), co uniemożliwiłoby znalezienie typu dla lewej części całego wyrażenia z powodu, cytuję "answer type incompatibility". Powyższe wyjaśnienie jest dość zdawkowe. Byłoby lepiej gdyby Autor pokazał szkielet drzewa dedukcji typu dla podanego wyrażenia (lub chociaż fragment tego drzewa) i na jego bazie wskazał, który warunek i dla jakiego podwyrażenia nie jest spełniony.
W kontekście podanego przez Autora przykładu, byłoby także interesujące pokazanie jak zachowuje się zaproponowany system typów dla języków $\lambda_{\mathcal{S}_0}$ i/lub $\lambda_{\$}$ w przypadku wyrażeń o których wiadomo, że nie są typowane używając klasycznej definicji systemu typów dla wariantów rachunku lambda. Przykładowo, wiadomo, że poniższe wyrażenie będące aplikacją funkcji:

$$(\lambda f.(f\ f))\ \lambda x.(\lambda y.x)$$

nie jest typowalne w ramach systemu typów języka ML. Było ciekawe przebadanie tego wyrażenia w podanym przez Autora systemie typów, rozszerzając powyższe wyrażenie o operator reset_0 i/lub \$.
11. Strona 24, drugie zdanie: "The type system satisfies the expected key properties such as subject reduction and progress, which is demonstrated next. Theorem 3.3 (...)". Z tytułu sekcji i jej początkowych zdań wynika, że Autor ma na myśli system typów $\lambda^{\leq}_{\mathcal{S}_0}$. Jakkolwiek dla wygody czytelnika należało podać w tym zdaniu nazwę systemu typów *explicite*, gdyż jak wspomniano wyżej, Figure 3.3 prezentuje system typów dla dwóch języków i nie jest wobec tego jasne czy Theorem 3.3 dotyczy języka $\lambda_{\mathcal{S}_0}$ czy $\lambda_{\$}$ (a może obu razem?).
12. Strona 24, Theorems 3.3 i 3.4: W porównaniu z resztą pracy, która reprezentuje wysoki poziom formalizacji, prezentacja dowodu poprawności systemu typów (*ang.* type soundness), na który składają się dowody twierdzenia 3.3 o zachowaniu typów (*ang.* type preservation lub subject reduction) oraz twierdzenia 3.4 o postępie (*ang.* progress), wydaje się być niepełna i mało rygorystyczna pod względem matematycznym, np.:

    - Podany dowód twierdzenia 3.3 jest w istocie szkicem dowodu: nie pokazano redukcji dla wszystkich konstrukcji języka oraz pominięto szereg lematów pośrednich

(zob. artykuł źródłowy [92] Wright'a i Felleisen'a oraz książkę [65] Bejnamin'a Pierce'a dla porównania). Byłoby więc lepiej zamiast słowa *Proof* użyć słowo *Proof sketch.* Analogiczna uwaga dotyczy także kilku innych dowodów, w tym szkiców dowodów zamieszczonych w głównej części rozprawy, dla których podano pełne dowody w załącznikach (Appendixes 1-5).

- Jeden z podanych wywodów cząstkowych "follows from a standard substitution lemma (that we omit)". Nie jest jasne dlaczego ten lemat jest pominięty — czy dlatego, że jest standardowy? O ile w artykułach naukowych często skraca się długie i żmudne dowody zachowania typów wyrażeń przez ich redukcję, to jednak w rozprawie doktorskiej, która ma pokazać opanowanie przez doktoranta naukowego warsztatu, wypadałoby dać ten klasyczny dowód bez skrótów dla całego języka lub chociaż dla wybranych, co ciekawszych konstrukcji tego języka. [1]
- Dowód twierdzenia 3.4 o postępie zawiera tylko stwierdzenie "Proof by straighforward induction on the structure of e". Jak już wspomniano wyżej, w rozprawie doktorskiej byłoby jednak dobrze pokazać przykładowe wywody dla co najmniej wybranych konstrukcji języka.

  Twierdzenie dotyczy wyrażeń zamkniętych (*ang.* closed terms) — dla pełności byłoby wskazane dodanie komentarza, że postęp nie jest (jest?) zachowany dla wyrażeń, które nie są zamknięte (tj. zawierają wolne zmienne).
- W wywodach obu dowodów brak jest odniesienia do konkretnych reguł typowania przez podanie ich etykiet (w końcu po to się je podaje w definicji systemu typów, aby przede wszystkim móc z nich korzystać w dowodach).
- W dowodzie twierdzenia 3.3 dla kompletności można było ustalić kilka technicznych lematów, np. wspomniany wcześniej kluczowy lemat o zastępowaniu, który pozwala na zastępowanie jednego podwyrażenia w typowanym wyrażeniu na inne podwyrażenie tego samego typu, bez wpływu na typ całego wyrażenia. Lemat o zastępowaniu jest konieczny aby móc pokazać zachowanie typu dla istotnego technicznego lematu o redukcji wyrażeń zawierających zamianę (*ang.* substitution) zmiennych na wartości. Tego typu lematy i dowody (lub fragmenty dowodów) powinny były znaleźć się w apendyksie.

13. Strona 27, sekcja 3.3.3, drugi paragraf: "Let us start by defining three families of mutually inductive predicates. The families are defined by induction on types: $\mathcal{R}_\tau$ is defined on well-typed values and is indexed by their types, $\mathcal{T}_\pi$ is defined on well-typed trails and is indexed by trail types, and finally $\mathcal{M}_\tau$ is defined on metacontexts and is indexed by their argument types. Trudno zrozumieć z tych definicji czym są $\mathcal{R}_\tau$, $\mathcal{T}_\pi$ i $\mathcal{M}_\tau$ — o jakie indukcyjne predykaty tutaj chodzi?
14. Strona 29, Lemma 3.16: "For every type τ, $\mathcal{T}_\tau$ (...)". Skoro przeważnie występuje $\mathcal{T}_\tau$, to

[1] Na marginesie warto przypomnieć POPLmark Challenge–niestety nieaktywny już projekt, którego celem była próba zautomatyzowania metateorii języków programowania, co pozwoliłoby usprawnić proces dowodzenia "standardowych twierdzeń".

dlaczego w definicji na stronie 27 napisano $\mathcal{T}_\pi$ (używając π, a nie τ), ale już w definicji $\mathcal{R}$ i $\mathcal{M}$ mamy τ?

15. Strona 29, Lemma 3.18: Co oznacza notacja w indeksach następujących dwóch rodzin predykatów: $\mathcal{T}_{\overline{\tau\sigma}}$ i $\mathcal{M}_{\downarrow\tau\sigma}$ i gdzie te symbole zostały zdefiniowane? (Jako komentarz należy w tym miejscu dodać, że czytając rozprawę daje się we znaki brak indeksu pojęć i oznaczeń.)
16. Strona 29, sekcja 3.3.4: "The type inference algorithm should reconstruct types for a given expression and find its principal type". W przypadku opracowania o charakterze monograficznym, jaką jest rozprawa doktorska, byłoby wskazane podać sformalizowaną definicję typu pryncypalnego lub chociaż refereencję do literatury, gdzie jest podana taka definicja, aby uniknąć nieścisłości. Przykładowo, w polimorficznym systemie typów ML definiuje się "pryncypalny schemat typu" korzystając z relacji generalizacji typów, gdzie "schemat typu" zawiera zmienne typu które mogą być zastępowane przez typy. W rozprawie jednak nie rozważa się polimorfizmu, więc o jaką generalizację chodzi?
17. Strona 32, sekcja 3.4, pierwsze zdanie: "The type system for $\mathsf{shift}_0/\mathsf{reset}_0$ introduced by Kiselyov and Shan [49] shares many properties with the type system presented in this work (..)". Czy ich system typów nie był zaproponowany wcześniej? Jeśli tak, to należało napisać: Our type system shares many properties with the type system introduced by Kiselyov and Shan [49]. Poza tym informacja, że już wcześniej zaproponowano (jakiś) system typów dla $\mathsf{shift}_0/\mathsf{reset}_0$ powinna była się pojawić w rozprawie dużo wcześniej, np. we wstępie opisującym motywacje, wskazując jednocześnie dlaczego system typów Kiselyov'a i Shan'a jest niedostateczny i stąd proponuje się nowy system typów dla tego języka (lub jego wariantu).
18. Strona 35, Figure 3.7: Czy zdefiniowano w tekście symbol redukcji $\Longrightarrow$?
19. Strona 37: ostatnie zdanie: "We can thus view this type system as an exotic type system for the shift operator (..)". Co znaczy "exotic type system" lub w jakim sensie jest egzotyczny?
20. Strona 41, sekcja 4.2, pierwsze zdanie: "The first CPS translation for $\mathsf{shift}_0/\mathsf{reset}_0$ in the literature was defined by Shan in [77]." Ta informacja powinna była pojawić się na początku rozprawy, w sekcji opisującej motywacje. Co nowego wnosi w tym względzie definicja CPS w prezentowanej rozprawie?
21. Strona 42, drugi i trzeci paragraf: zamiast "the Shan's translation" powinno być: Shan's translation (bez "the").
22. Strona 56, odnośnik 2: "Maciej's talk cannot be properly referenced". W praktyce spotyka się czasem referencje opisane jako "personal communication" i taka powinna być umieszczona w bibliografii dla jej kompletności.
23. Strona 65, sekcja 6.2.1, pierwsze zdanie: "The CPS hierarchy with $shift_0/dollar$ (...)". Powinno być: the CPS hierarchy with $\mathsf{shift}_0/\$$ (...).
24. Strona 70, sekcja 6.3.1, pierwsze zdanie "(..) one of the classic applications of the CPS hierarchy is an implementation of McCarthy's *amb* non-deterministic choice operator (..)". Brak odnośnika do pracy McCarthy'iego.

25. Strona 75, Property 7.1: Czy w regułach 3 i 4 nie powinno być samo λ zamiast λ_{S_0} i $\lambda_{\$}$? (Obecnie występuje niespójność symboli w regule i jej tekstowym wprowadzeniu.)
26. Strona 81, drugi paragraf: "for impure application". Powinno (raczej) być: for a/the impure application.
27. Strona 91, pierwszy paragraf: "(...) the type-directed CPS translation, which transforms selectively fragments of the program source, requires implementing in the compiler the type system presented in Chapter 3, which, if our goal is to implement $\mathsf{shift}_0/\mathsf{reset}_0$ in a existing, practical programming language, is a hard task [72]". To zdanie jest zagmatwane i przez to niejasne. Translacja CPS ma tradycyjnie zastosowanie właśnie w kompilatorach. Wydaje się więc, że Autorowi chodziło raczej o uzasadnienie, że nie zaimplementował proponowanych mechanizmów wewnątrz kompilatorów znanych języków programowania, gdyż byłoby to po prostu zbyt trudne i pracochłonne zadanie (co jest prawdą). Jakkolwiek na marginesie warto dodać, że jako *proof-of-concept* wystarczyłoby zaimplementować kompilator i maszynę abstrakcyjną dla prostego rachunku lambda, rozszerzonego o konstrukcje $\mathsf{shift}_0/\mathsf{reset}_0$, co byłoby w pełni wykonalne w ramach doktoratu. Z drugiej strony rozprawa ma charakter teoretyczny, wobec tego implementacja nie wydaje się być warunkiem koniecznym pozytywnej oceny rozprawy. W tym kontekście, podjęcie się przez Autora implementacji w formie biblioteki dla C/C++, opisanej w rozdziale 8, należy uznać za duży plus, gdyż pokazanie praktycznego zastosowania wyników teoretycznych wzmacnia dodatkowo celowość podjęcia się tych badań w dyscyplinie naukowej jaką jest informatyka.
28. Strona 91, ostatnie zdanie: "C++11". Brak referencji do tego nowego standardu C++.
29. Strona 101, sekcja 8.3.2, tytuł: "Lightweight threads". Bardziej adekwatny do treści tytuł byłby np. "multithreaded server".
30. Strona 103, Conclusions, pierwsze zdanie: "In this thesis we developed many important theoretical results (...)". Lepiej: In this thesis we developed many novel theoretical results, zostawiając ocenę na ile prezentowane rezultaty są rzeczywiście ważne tym, którzy zechcą je cytować.

Wyżej wymienione uchybienia i nieścisłości są drobne, za wyłączeniem tych wymienionych w punktach 9-12, tj. odnoszących się do ekspresywności zaproponowanego systemu typów i dowodu jego poprawności (Theorem 3.3 i 3.4). Warto też dodać na marginesie, ze system typów jest dość prosty i nie uwzględnia polimorfizmu. Jakkolwiek w konkluzjach Autor wskazuje opracowanie polimorficznego systemu typów dla swoich operatorów jako możliwą kontynuację badań. Potencjalnie istnieją też inne możliwości kontynuowania przedstawionych w rozprawie badań.

Dyskusyjne jest dlaczego w pracy dotyczącej abstrakcji sterowania dla kontynuacji nie próbowano dodać do języka konstrukcji które manipulują stan, np. referencje w stylu ML. Przydałby się w rozprawie jakiś drobny komentarz w tym względzie. Czy rozszerzenie języka o referencje niczego ciekawego by nie wniosło do badania kontynuacji? Czy może wręcz odwrotnie, ale rozszerzenie języka o te dodatkowe konstrukcje nie było brane pod uwagę z po-

wodu ograniczeń czasowych lub innych przyczyn? Uwzględnienie referencji skomplikowałoby bowiem meta-teorię, np. wymagałoby rozszerzenia zarówno semantyki języka jak i systemu typów o magazyn danych (*ang.* data store) i jego typowanie.

Od strony redakcyjnej, istotnym niedociągnięciem Autora jest brak w rozprawie indeksu używanych symboli i oznaczeń, co dużej mierze utrudnia zrozumienie rozprawy.

W tym miejscu należy jednak podkreślić, że przedstawione wyżej uwagi nie wpływają istotnie na całościową ocenę przedstawionej pracy, którą oceniam wysoko.

Reasumując stwierdzam, że cel rozprawy doktorskiej został w pełni osiągnięty, a jej główna teza została pozytywnie zweryfikowana. Moja ocena rozprawy pod względem trzech podstawowych kryteriów jest następująca:

A. Czy rozprawa zawiera oryginalne rozwiązanie problemu naukowego?

- *Zdecydowanie TAK* **X**
- *Raczej TAK*
- *Trudno powiedzieć*
- *Raczej NIE*
- *Zdecydowanie NIE*

B. Czy po przeczytaniu rozprawy zgadzasz się, że kandydat posiada dostateczną wiedzę w dyscyplinie Informatyka?

- *Zdecydowanie TAK* **X**
- *Raczej TAK*
- *Trudno powiedzieć*
- *Raczej NIE*
- *Zdecydowanie NIE*

C. Czy kandydat posiada umiejętność samodzielnego prowadzenia pracy naukowej?

- *Zdecydowanie TAK* **X**
- *Raczej TAK*
- *Trudno powiedzieć*
- *Raczej NIE*
- *Zdecydowanie NIE*

W podsumowaniu stwierdzam, że przedstawiona rozprawa doktorska mgra Materzoka wnosi ważny i wartościowy wkład do aktualnej problematyki badawczej w obszarze semantyki języków programowania. Doktorant wykazał się dobrą znajomością wysoce specjalistycznej dziedziny, którą się zajął, potrafi formułować oryginalne i ambitne problemy naukowe oraz rozwiązywać je zarówno w dziedzinie teoretycznej jak również implementacyjnej. Wymienione elementy upoważniają do jednoznacznego stwierdzenia, że spełnione są wymagania *ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki* (Dz. U. z 2003 r. Nr 65, poz. 595 z późniejszymi zmianami) związane z uzyskaniem stopnia doktora nauk technicznych w dyscyplinie Informatyka. W związku z

powyższym **stawiam wniosek o dopuszczenie mgra Marka Materzoka do dalszych etapów postępowania w celu uzyskania stopnia naukowego doktora.**

Ponadto, biorąc pod uwagę wysokie oceny cząstkowe, rekomenduję wyróżnienie przedstawionej rozprawy doktorskiej.

Podpis