

# Tomasz Gogacz

September 22, 2016

data urodzenia: 01.06.1987  
e-mail: tgogacz@gmail.com

## 1 Przebieg kariery zawodowej

- Staże na Uniwersytecie Warszawskim, Opiekun: Mikołaj Bojańczyk  
09.2012 - 03.2013 (funded by “National PhD program in Mathematics and Computer Science”)  
02.2014 - 06.2014 (funded by “Warsaw Center of Mathematics and Computer Science”)
- Postdoc na Uniwersytecie W Edynburgu 10.2015 - 6.2016
- **Obecne zatrudnienie:** Adiunkt na Uniwersytecie Warszawskim 9.2016-8.2017

### 1.1 Edukacja

- Studia magisterskie na kierunku Informatyka (2006-2011), Instytut informatyki na Uniwersytecie Wrocławskim  
Promotor: Jerzy Marcinkowski
  - średnia ocen: 4.96
  - ECTS credits: 326
- Studia magisterskie na kierunku Matematyka (2007-2011), Instytut matematyczny na Uniwersytecie Wrocławskim  
Promotor: Krzysztof Krupiński
  - średnia ocen: 4.94
  - ECTS credits: 376
- Studia Doktoranckie (2011- 2016), Instytut informatyki na Uniwersytecie Wrocławskim  
Promotor: Jerzy Marcinkowski
  - data obrony: 05.04.2016
  - Tytuł rozprawy: “Tuple generating dependencies i związki między nimi”

## 2 Publikacje:

- On the BDD/FC Conjecture  
PODS 2013, co-author: Jerzy Marcinkowski
- Converging to the Chase – a Tool for Finite Controllability,  
LICS 2013, co-author: Jerzy Marcinkowski
- On regular groups and fields,  
Journal of Symbolic Logic, co-author: Krzysztof Krupiński (this is my Masters’ of Mathematics Thesis)
- On the decidability of MSO+U on infinite trees  
ICALP 2014, co-authors: Mikołaj Bojańczyk, Henryk Michalewski, Michał Skrzypczak
- Measure Properties of Game Tree Languages  
MFCS 2014, co-authors: Henryk Michalewski, Matteo Mio, Michał Skrzypczak
- All-instances termination of chase is undecidable  
ICALP 2014, co-author: Jerzy Marcinkowski

- The Hunt for a Red Spider: Conjunctive Query Determinacy Is Undecidable  
LICS 2015, co-author: Jerzy Marcinkowski
- Non-dominating Sequences of Vectors Using only Resets and Increments  
Fundamenta Informaticae 2015, co-authors: Wojciech Czerwiński, Eryk Kopczyński
- Red Spider Meets a Rainworm: Conjunctive Query Finite Determinacy Is Undecidable  
PODS 2016, co-author: Jerzy Marcinkowski
- Entropy bounds for conjunctive queries with functional dependencies  
ICDT 2017, co-author: Szymon Toruńczyk

### 3 Nagrody i Stypendia

- 2014 Scholarship for best PhD students in department
- 2013 Scholarship for best PhD students in department
- 2013 Rector's scholarship for best PhD students
- 2012 Scholarship for best PhD students in department
- 2012 Rector's scholarship for best PhD students
- 2011 Scholarship for best PhD students in department
- 2011 2nd prize in competition "Józef Marcinkiewicz" for the best student paper in mathematics.
- 2011-2015 Scholarship of "National PhD program in Mathematics and Computer Science".
- 2010-2011 Scholarship of the Minister of Science and Higher Education.
- 2009-2010 Scholarship of the Minister of Science and Higher Education.

### 4 Granty

2014-2016 The BDD/FC Conjecture and beyond,

Research grant funded by the National Center of Science, principal investigator.

2013 Decidability of query rewriting over views,

Research project 1398/M/IIn/13, funded by the University of Wrocław, principal investigator.

2012 Finite Controllability for some extension of Datalog,

Research project 2159/M/IIn/12, funded by the University of Wrocław, principal investigator.

#### 4.1 Hobby

- Drugie miejsce w drużynowych mistrzostwach polski młodzieży w brydżu, 2006
- Japoński

## 5 Opis dotychczasowej pracy naukowej

Na mój doktorat składały się cztery prace badające pewien formalizm reprezentacji baz danych, o których wiemy że są niepełne. Na przykład gdy konstruujemy swoje drzewo genealogiczne możemy nie znać imion wszystkich możliwych przodków. Możemy za to bezpiecznie założyć, że każdy z tych przodków miał matkę. Tego typu wiedzę reprezentujemy za pomocą zdań w logice pierwszego rzędu zwanych “Tuple generating dependencies” (w skrócie TGD). Mają one postać

$$\forall \bar{x}, \bar{y} \quad \Psi(\bar{x}, \bar{y}) \Rightarrow \exists \bar{z} \quad \Phi(\bar{z}, \bar{y})$$

, gdzie  $\Phi$  i  $\Psi$  są zapytaniami koniunkcyjnymi (są to najprostsze możliwe zapytania do baz danych mające formę koniunkcji warunków).

Dwie pierwsze prace dowodziły, że pewne klasy TGD mają własność Finite Controllability (FC). Jest to pewien wariant własności modelu skończonego, mówiący że formuła ma model wtedy i tylko wtedy gdy ma model skończony. Jest to własność istotna z następującego powodu. Chcialibyśmy wnioskować (odpowiadać na zapytania) o naszej “idealnej” bazie danych, to znaczy o tej która zawiera wszystkie interesujące nas informacje. Siłą rzeczy bazy danych są skończone, więc interesują nas skończone modele formuł. Z drugiej strony Wnioskowanie w logice staje się dużo łatwiejsze, gdy dopuszczamy możliwość istnienia nieskończonych struktur (co mogłoby prowadzić do niedokońca prawdziwych wniosków). Własność FC mówi że oba podejścia są równoważne, co daje względnie łatwy sposób wnioskowania o skończonych strukturach.

Kolejna praca dotyczyła modelowania migracji między bazami danych gdy TGD jest używany jako język specyfikacji. Język ten jest na tyle ekspresywny, że można w nim wyrazić w sposób deklaratywny (czyli bez szczegółów implementacyjnych) prawie wszystkie standardowe sytuacje. Niestety, jak to zwykle bywa z ekspresywnymi formalizmami, pojawiają się pytania czy konkretna specyfikacja “ma sens”. Problem rozważany w tej pracy skupiał się na fundamentalnym pytaniu, czy dla danej specyfikacji automatyczna migracja danych kierowana tą specyfikacją zakończy swoje działanie w skończonym czasie. Okazuje się, że ten problem jest nierozstrzygalny, co znaczy że ta kwestia musi być rozważona przez programistów.

Ostatnia praca w doktoracie dotyczyła Problemu determinacji zapytań koniunkcyjnych. W tym problemie pytamy się czy odpowiedzi na wcześniejsze zapytania  $Q_1, \dots, Q_n$  zawierają koniecznie dosyć informacji aby odpowiedzieć na kolejne zapytanie  $Q$ . Jeśli tak jest, to jesteśmy w stanie obliczyć odpowiedź na  $Q$  bez zaglądania do oryginalnej bazy danych: wystarczy nam odpowiedzi do zapytań  $Q_1, \dots, Q_n$  które są dostępne lokalnie. Okazuje się, że również ten problem jest nierozstrzygalny. Ciekawostką jest fakt, że w sformułowaniu problemu nie pojawiają się TGD. Są one jednak bardzo istotnym narzędziem, których użycie pozwoliło na rozwiązanie tego problemu.