Danil Annenkov
4 Allée Paul Le Drogo, Nantes, 44300, France
+33 755 361 517
danil.v.annenkov@gmail.com, danil.annenkov@inria.fr
Personal page: dannenkov.me

EDUCATION

PhD, University of Copenhagen, DIKU — November, 2014 - October, 2017

Graduated with distinction from Irkutsk State Technical University,
Faculty of Cybernetics, Department of Automated Systems (5 of 5 years)
(equivalent to a master's degree) — July, 2002

RESEARCH INTERESTS

Formal semantics of programming languages, functional programming, certified programming, type theory, homotopy type theory, applications of category theory in programming language research.

PUBLICATIONS AND PREPRINTS

1. Danil Annenkov, Martin Elsman. Certified Compilation of Financial Contracts. Under review PPDP'19.
2. Martin Elsman, Troels Henriksen, Danil Annenkov, Cosmin Oancea. Static Interpretation of Higher-Order Modules in Futhark, accepted at ICFP'18.
3. Danil Annenkov, Paolo Capriotti, Nicolai Kraus. Two-Level Type Theory and Applications. Submitted to ACM Transactions on Computational Logic (TOCL), May 2017.
   [arXiv e-print: https://arxiv.org/abs/1705.03307].
4. Danil Annenkov and Martin Elsman. Towards Certified Compilation of Financial Contracts. In *Proceedings of the 28th Nordic Workshop on Programming Theory (NWPT'16)*. Aalborg, DK. November 2016.
   [http://dannenkov.me/papers/NWPTPayoffLang.pdf]
5. Annenkov, D.V. , Cherkashin, E.A., Generation technique for Django MVC web framework using the Stratego transformation language. *36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, pp. 1084 -1087, 2013.
6. Annenkov, D.V., Domain-specific languages as a foundation of the approach to software development. *Proceedings of conference with international participation "Vinerovskie chteniya"*, pp. 142-146 (in Russian), 2013.
7. Evgeny A. Cherkashin , Polina V. Belykh, Danil V. Annenkov, Christina K. Paskal A Document Content Logical Layer Induction on the Base of Ontologies and Processing Changes. *Proceedings of International Conference on Applied Internet and Information Technologies*, University of Novi Sad Technical Faculty «Mihailo Pupin», Zrenjanin, Serbia, 2013. C. 252-257.
8. Cherkashin, E.A., Paramonov, V.V., Fedorov, R.K., Terehin, I.N., Pozdnyak, E.I., Annenkov, D.V., Information Systems Framework Synthesis on the Base of a Logical Approach.

Proceedings of International Conference on Applied Internet and Information Technologies, October 26, Serbia, Zrenjanin. pp. 239-244, 2012.
9. Annenkov, D.V., Simulation and research of the securities market. *Bulletin of the Irkutsk State Technical University*, vol. 27, #3, pp. 68-71 (in Russian), 2006.
10. Annenkov, D.V., Petukhov, P.A., Torbeeva, A.S., Modeling the organization and functioning of the securities market, *Proceedings of conference "Vinerovskie chteniya"*, pp. 35-40, 2005.

PRESENTATIONS

1. *Nominal Techniques in Coq.* HIPERFIT Workshop 2018, University of Copenhagen, November 2017.
2. Danil Annenkov (work with Paolo Capriotti and Nicolai Kraus). Formalisations Using Two-Level Type Theory. HoTT/UF Workshop 2017, Oxford, UK. September 2017.
3. *Towards Certified Compilation of Financial Contracts. Nordic Workshop on Programming Theory (NWPT'16)*. Aalborg, DK. November, 2016.
4. *Verifying the generation of payoff-language expressions.* HIPERFIT Workshop, Copenhagen, 2016
5. *Domain-specific languages as a foundation of an approach to the software development.* Conference with international participation "Vinerovskie chteniya", Irkutsk State Technical University, Russia, 2013
6. Annenkov, D.V., *Simulation as a set of methods and tools for analysis and synthesis of various systems (on the example of the securities market)*. Conference with international participation "Vinerovskie chteniya", Irkutsk State Technical University, Russia, 2005.

GRADUATE LEVEL COURSES

1. Semantics and Types.
   Lecturer: Andrzej Filinski, Associate Professor at Department of Computer Science, University of Copenhagen.
2. Static Program Analysis and Language-based Security.
   Lecturer: Thomas Jensen, Directeur de recherche INRIA, Rennes.

SUMMER SCHOOLS, SEMINARS

1. Type theory seminar at IT University of Copenhagen organised by Rasmus Møgelberg, February - May 2017.
   Reading papers and discussing such topics as categorical models of type theory, type theory with guarded recursion, homotopy type theory. I contributed with a talk on definitions of equivalence in homotopy type theory.
2. Midlands Graduate School in the Foundations of Computing Science 2017, University of Leicester, Leicester, UK.
   Following "Advanced Courses": higher-category theory, graphical linear algebra, coalgebras and Infinite Data Structures
3. Blockchain Summer School 2016, IT University of Copenhagen, Copenhagen, Denmark.
   Introduction to blockchain technology, distributed ledgers with hand-on experience with the

Ethereum platform and Solidity smart contract programming language.

4. Midlands Graduate School in the Foundations of Computing Science 2016, University of Birmingham, Birmingham, UK.
   Following “Basic Courses”: type theory, category theory, denotational semantics.
5. Oregon Programming Language Summer School 2015, University of Oregon, Eugene, US.
6. Reading group on data-parallel programming languages, 2015 -2016.
   Discussion of papers related to type systems, cost semantics, implementation and application of data-parallel programming languages.

ACADEMIC/TEACHING EXPERIENCE

Postdoc, INRIA Nantes Since February, 2018

Teaching assistant, co-teacher 2015-2017

University of Copenhagen, DIKU.
Courses:
Advanced Java (concurrency, java generics, testing, RPC over HTTP, working with databases)
Advanced Computer Systems (concurrency control - serializability, two-phase locking; database recovery protocols, consensus protocols in distributed systems).
Co-supervision of Bachelor’s projects

Part-time Lecturer September, 2010 - 2012

Irkutsk State Technical University, Faculty of Cybernetics
Taught the following courses: Web Programming, Technology of Programming.

Teacher July, 2010 - March, 2011

International Education Centre Aptech-ISTU

- Java Programming Language (ACCP Russia, semester 2) course.

Graduate Assistant September, 2002 - 2007

Irkutsk State Technical University, Faculty of Cybernetics

- Courses taught: Algorithmic Programming Languages, Organization of Databases, Operating Systems.
- Developed new curriculum of “Technology of Programming” course.

Graduate Research Assistant September, 2002 - 2007

Irkutsk State Technical University, Faculty of Cybernetics

- Stochastic modelling of financial time series.
- Development of tools for generation of random numbers with given properties.

FACULTY ACTIVITIES

Organizing committee member of the annual regional programming contest 2003 - 2005
Organizing committee member of the conference "Vinerovskie chteniya" 2004

PROFESSIONAL EXPERIENCE

Founder of Software Development Company 2009 - 2014

Web-based enterprise solutions
(CRM, Service Department Information System, e-Commerce)

Python/Javascript Developer September 2013 - January 2014

Exploriana Inc.

Member of guidemore.com development team.
Guidemore.com is a internet service for creation travel guides for mobile devices.

Programmer 2004 - 2005
Railroad Clinical Hospital, Irkutsk, Russia
Deployment of a medical information system

SKILLS

10+ years of programming experience (both industry and academic).

Programming languages
Java, Python, Javascript, Scala, Haskell, Racket (Lisp dialect), Stratego/XT.

Proof assistants
Coq, Lean, Agda (some experience)

Markup languages
XML, HTML, Markdown, reStructuredText

Version control systems
Svn, Git, Mercurial

Team Leadership
Led team of developers (3-4 persons).

LANGUAGES

Russian - native
English - advanced, IELTS score 7.0 in 2014.

PROJECTS

1. **Two-level type theory**

   An implementation of two-level type theory in the Lean proof assistant. Two-level type theory is a version of Martin-Löf type theory (MLTT), which consists of two fragments: the fibrant fragment (Homotopy Type Theory) and the strict fragment (MLTT with uniqueness of identity proofs).
   https://github.com/annenkov/two-level
2. **Contract DSL**
   Some extensions of the original Contract DSL (Patrick Bahr, Jost Berthold, Martin Elsman. Certified Symbolic Management of Financial Multi-Party Contracts, ICFP'2015) and compilation to the intermediate Payoff language implemented in Coq proof assistant.
   https://github.com/annenkov/contracts
3. **Simply-Typed Lambda-calculus in Coq**
   The projects includes:
   - The proof of normalization of the Call-by-Value Simply-Typed Lambda Calculus (STLC) using Tait's method in Coq.
   - Implementation of nominal sets with applications to STLC (nominal set of lambda terms, definition of alpha-equivalence, proof of equivariance of the typing relation)

   https://github.com/annenkov/stlcnorm

4. **The HIPERFIT Portfolio Management Prototype**
   Web-based system that integrates the Contract DSL and the HIPERFIT parallel pricing engine (FINPAR). The system is written in Haskell and features automatic web-form generation on base of Haskell data types along with OpenCL code generation for payoff functions.
   https://github.com/HIPERFIT/prototype
5. **Stratego-SLL**
   An interpreter and minimal positive supercompiler for first-order functional language (M_0 language from the Morten Sørensen's Master Thesis) written in Stratego transformation language (part of the Spoofax language workbench).
   https://github.com/annenkov/stratego-sll
6. **Unmix project ported to Racket**
   Unmix was originally developed by Sergei A. Romanenko (Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, Moscow, Russia)
   Unmix is a program specializer (based on partial evaluation) for a subset of Scheme.
   https://github.com/annenkov/unmix
7. **Entity Model DSL**
   Simple textual DSL for modelling entities with Python code generation.
   https://bitbucket.org/ib_soft/entity-model/
8. **Some bioinformatics algorithms in Haskell**
   Implementation of brute-force Regulatory Motifs Finding algorithm (formulated as a Median String problem) with two branch-and-bound strategies.
   https://bitbucket.org/ib_soft/haskell-samples