Nikita Frolov

nf@mkmks.org
github.com/mkmks
linkedin.com/in/mkmks
+46 73 571 39 89

Summary

I am an apprentice researcher in the area of programming language design. The bulk of ideas fundamental to my work comes from type theory and category theory. I am also versed in computer architecture, information security and human linguistics. I seek a position in industry where my academic background can be an advantage.

Employment

2012-now **PhD student, Functional Programming Group**
Chalmers University of Technology, Gothenburg, Sweden
Funded by the Resource-Aware Functional Programming project (RAWFP)
Thesis advisor: John Hughes

My research is aimed at giving the functional programmer the control over low-level aspects of execution, so writing functional programs for embedded hardware becomes a reality. In particular, I work on bringing the well-known *tiling optimizations* to the category theory-inspired *recursion schemes.* To formalise the connection between tiled and untiled recursive programs, I rely on the concepts of datatype *ornaments* and *transporting* functions across ornaments.

I have completed the requirements for the *licentiate* degree. I have also completed the graduate course requirements and the teaching duties of my doctoral program. I expect to graduate by the end of 2017.

2011 **Research assistant, Language Technology Group**
Chalmers University of Technology, Gothenburg, Sweden
Funded by the Multilingual Online Translation project (MOLTO)
Project leader: Aarne Ranta

I have implemented a number of improvements for the Russian resource grammar in the Grammatical Framework, including support for partitives and ordinals.

2009 **Freelance developer**

I have implemented a connector between the OpenCA phone softswitch and SORM, the Russian lawful interception system. The connector would work by running a man-in-the-middle attack on the user traffic. (Python, Twisted, SIP/RFC3261, RTP/RFC3550)

2005-2009 **Developer**
Demos Company Ltd., Moscow, Russia

- I have maintained a custom SIM/USIM-card lifecycle management solution for a large mobile operator (more than 100M active subscribers).
(Spring, Hibernate)
- I have prototyped a number of applications for the nCipher/Thales hardware security modules: SIM/USIM-card key generation and storage procedures, GOST cryptographic algorithms, a host-side PKCS11 API wrapper (C)
- I have extended the RSA MobileID authentication server with the support for GOST algorithms and a CT-KIP implementation. (Java)

Education

2012-2016 **Licentiate in Computer Science**
Chalmers University of Technology, Gothenburg, Sweden

A licentiate degree is awarded in Sweden and Finland for a dissertation that is roughly one half of a doctoral dissertation. In my licentiate dissertation, I have built a type-theoretical model of *tiled container traversals*. Tiling is an optimization aimed at improving locality of memory accesses. I have demonstrated that this optimization can be beneficial for functional programs by implementing it as a part of a compiler for a simple functional language.

2009-2011 **M.Sc. in Computer Engineering**
Chalmers University of Technology, Gothenburg, Sweden

My thesis work presented an approach to the *phase sequencing problem* based on expressing the scheduling problem as a set of mutually independent constraints and solving them with a SAT solver. To evaluate the approach, I created a compiler for FlexCore, an embedded processor with reconfigurable architecture. The compiler was built on top of a scheduling constraints DSL embedded into Haskell.

2003-2008 **B.Sc. in Electrical Engineering**
Bauman Moscow State Technical University, Moscow, Russia

Publications

2016 **Laying Tiles Ornamentally: An approach to structuring container traversals**
N. Frolov
Licentiate thesis
Department of Computer Science and Engineering
Chalmers University of Technology and Gteborg University
ISSN 1652-876X, No. 161L, Dec. 2016
http://publications.lib.chalmers.se/publication/246117

2011 **A SAT-based Compiler for FlexCore**
N. Frolov, M. Sjlander, P. Larsson-Edefors, S. A. McKee
Technical report
Department of Computer Science and Engineering
Chalmers University of Technology and Gteborg University
ISSN 1652-926X, No. 11-04, Apr. 2011
http://publications.lib.chalmers.se/publication/143872-a-sat-based-compiler-for-flexcore

Talks

2016 **Laying Tiles Ornamentally**
28th Symposium on Implementation and Application of Functional Languages (IFL)
August 31, 2016, KU Leuven, Belgium

2015 **Decomposing Schedules of Recursive Programs**
13th Symposium on Trends in Functional Programming (TFP)
June 4, 2015, INRIA Sophia Antipolis, France

2011 **Declarative, SAT-solver-based Scheduling**
for an Embedded Architecture with a Flexible Datapath
11th Swedish System-on-Chip Conference (SSoCC)
May 2, 2011, Varberg, Sweden

Teaching

I have been a teaching assistant responsible for conducting practice sessions and grading weekly assignments and final exams in several courses given as part of B.Sc. and M.Sc. programs in computer science at Chalmers:

- Parallel functional programming (2012-2016, M.Sc.-level)
- Algorithms (2015-2016, M.Sc.-level)
- Cryptography (2013-2015, M.Sc.-level)
- Programming paradigms (2012-2015, B.Sc.-level)
- Data structures (2012, B.Sc.-level)

I have developed and delivered two lectures for the parallel functional programming course: an introduction to the GHC runtime system (thread scheduling, garbage collection, IO) and an introduction to cache hierarchies (general concepts of CPU caches, cache-aware and cache-oblivious algorithms). I have given the former lecture during three instances (2014-2016) and the latter during four instances of the course (2013-2016).

Skills

My day-to-day tools are the **Agda** proof assistant and the LaTeX document preparation system. To produce programs that can not only be typechecked and typeset but also benchmarked, I use **Haskell** (in its GHC variety) and **C**.

I remember enough **VHDL** to appreciate the existence of languages that compile into it. I remember enough **Java** to teach the standard data structures and algorithms courses. I have struggled enough with **Python** to wish never to implement network protocols in an untyped language again.

Other programming languages I have had encounters with include **Coq**, **Scala**, **C++**, **Emacs Lisp** and the **Nix** package description language. I have not done any major work in those though.

Other

I am a Swedish and a Russian citizen.

Besides being fluent in English, Swedish and Russian, I can get by in German and French.