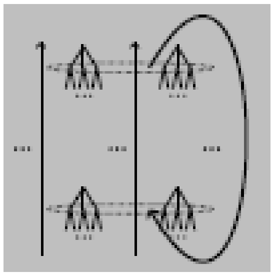


MICHAEL A. BUKATIN

=====

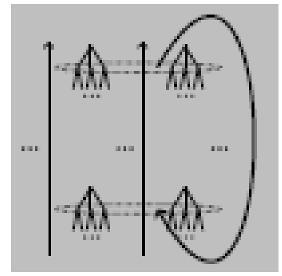
P.O.Box 391894, Cambridge, MA 02139

e-mail: bukatin@cs.brandeis.edu

=====

Research Scientist

Software Engineer



SUMMARY: Extensive experience as a software engineer, PhD in computer science, authorship of results in the field of mathematics instrumental in the theory of programming languages, strong background in mathematics and computer science, familiarity with theoretical neuroscience, consulting and teaching experience.

Strong interest in methods and tools for machine learning, various flavors of neural networks, attention-based models including Transformers, probabilistic programming, program synthesis, evolutionary algorithms, and computer art. **Dataflow Matrix Machines** as generalized neural networks and as a programming framework.

Design of programming languages, design and implementation of compilers, search engine indices, search quality metrics, advanced algorithms, computer graphics, user interfaces, small editors, computer algebra and differentiable programming, sparse matrices and tensors, numerical and statistical methods.

RESEARCH FOCUS: Continuous models of computations (various flavors).

KEY RESEARCH DISCOVERY: **Dataflow Matrix Machines**, a novel class of neural machines.

- a highly expressive generalization of self-modifying neural networks;
- a general-purpose programming platform based on neuromorphic computations with linear streams;
- dataflow programming in the style of composition of unit generators for digital audio synthesis;
- programs are equivalent to connectivity matrices and can be continuously deformed;
- use of **flexible tensors** based on tree-shaped indices.

We expect that further development of Dataflow Matrix Machines will lead to progress in programs synthesis, methods for learning to learn, neuro-symbolic architectures, hierarchical learning, and more. Separate elements of this approach can be fruitfully used in a variety of existing contexts. There is a lot of affinity between linear streams and attention mechanisms used in Transformers. Modern differentiable programming frameworks such as Zygote.jl (Julia Flux) and JAX are shown to successfully handle flexible tensors based on tree-shaped indices.

MAIN OBJECTIVES

- To use attractive theoretical properties of Dataflow Matrix Machines in practical applications.
- To use Dataflow Matrix Machines to enhance the magic of modern Transformers.
- To advance research on existential safety of future AI systems.

ENGINEERING TOOLS

OPERATING SYSTEMS: **Linux, Windows,** other flavors of UNIX, other operating systems.

PROGRAMMING LANGUAGES: **Julia*, Python*, Clojure, C, Processing, GLSL, Java, C++,** JavaScript, Fortran, R, SQL, Common Lisp, Pascal, Scheme, ML, Prolog, (Perl, Ada, Basic, PL/1, Algol-60, Modula-2).

MACHINE LEARNING FRAMEWORKS: **Julia Flux/Zygote*, JAX, PyTorch,** TensorFlow, DyNet.

LARGE LANGUAGE MODELS: GPT-4 and its relatives.

EXPERIENCE

01/2022-09/2022: Pasteur Labs, Inc., remote work — Senior Research Scientist.

Advanced ML and AI methods and architectures; applications to modeling, simulation, and natural sciences. Julia, Python, Julia Flux, JAX, Linux workstations.

11/2018-09/2019: HERE North America, LLC., Burlington, MA — Lead ML & AI Engineer.
12/2012-11/2018: HERE North America, LLC., Cambridge/Burlington, MA — Senior Software Engineer.
04/2010-12/2015: Nokia Corp., Boston/Cambridge/Burlington, MA — Senior Software Engineer.

Design and development of *search quality metrics*, *ranking models*, and related software for local search in Nokia Maps (HERE Maps) in collaboration with other members of Nokia (HERE) engineering team.
Python, C++, R, Java, PostgreSQL, Clojure, PyTorch, TensorFlow, DyNet, Linux workstations.

07/2001-04/2010: MetaCarta, Inc., Cambridge, MA — Senior Software Engineer.
06/2001-07/2001: MetaCarta, Inc., Cambridge, MA — Consultant.

Design and development of efficient algorithms and software for *feature extraction* and *indexing* in the **GTS** (Geographic Text Search) Appliance in collaboration with other members of MetaCarta engineering team.
C, C++, FLEX, Python, PostgreSQL, Linux workstations.

06/1998-10/2000: Synquiry Technologies, Ltd., Belmont, MA — Consultant.

Design of a modern *agent-oriented programming language* and *semantic-oriented visual program editing system* for development of AI agents working over Ariadne semantic networks (models).
Common Lisp, Sun and Windows NT workstations.

03/1995-06/1997: Parametric Technology Corp., Waltham/Newton, MA — Senior Software Engineer.

Design and implementation of *application oriented programming languages.* Design and implementation of prototype and industrial systems (CAD/CAM) based around these languages.
Data flow model of computations, incremental compilation and computations.
C++, YACC, LEX, proprietary application oriented languages. UNIX and Windows NT workstations.

07/1990-11/1992: Biosym Technologies, Inc., Parsippany, NJ — Scientific Programmer.
04/1990-06/1990: Mount Sinai Medical Center, New York, NY — Research Assistant.

Design, development and use of software computing hydration energy of molecules. DAMS software project: construction of divided analytical molecular surface known as water accessible Richards-Connolly surface. Advanced computational geometry algorithms, advanced computer graphics. C, Fortran. SGI workstations, VAX, Convex. 5 publications in *Biophysical Chemistry* and *The Journal of Physical Chemistry.*

09/1986-08/1989: Institute of Economics, Moscow, USSR — Researcher.

Various software design and development. Large computer algebra and differentiable programming project (exact derivatives of functions defined by Fortran programs). Support of end users. Pascal, Fortran, C. IBM PC, Nord (VAX-like platform). From 12/88 on consulting basis.

RESEARCH, TEACHING, AND EDUCATION

Fall 2012-present: Research collaboration on **Vector Semantics of Programming Languages**.

In the context of several academic research collaborations, I have been leading the efforts related to vector semantics of programming languages and to *programming with linear streams* (data streams equipped with the ability to take linear combinations of several streams).

- **2012-2013:** discovery of deep connections between partial inconsistency and vector semantics of programming languages, leading to the discovery of *partial inconsitency landscape*;
- **2014-2015:** introduction and study of the discipline of dataflow programming with linear streams; first open-source prototypes; the discovery of *Dataflow Matrix Machines*;
- **2016-2019:** study of Dataflow Matrix Machines as generalized neural networks and as a programming platform; discovery of V-values (*flexible tensors* based on tree-shaped indices); open source research-grade implementations.
- **2020-present:** study of interplay between Dataflow Matrix Machines and attention-based models including Transformers; *neural machines and properties of matrix multiplication*; Dataflow Matrix Machines and modern differentiable programming (Julia Flux/Zygote.jl, JAX); *novel methods for neural architecture search and program synthesis*; invariant properties of self-modifying systems; modulating fields.

Dataflow Matrix Machines links:

GitHub Pages: `https://anhinga.github.io`
Open source implementation (Clojure): `https://github.com/jsa-aerial/DMM`
Two-page white paper: `https://anhinga.github.io/brandeis-mirror/dmm-white-paper-2024.pdf`
Research agenda: `https://anhinga.github.io/brandeis-mirror/dmm-collaborative-research-agenda.pdf`
Reference slide deck: `https://github.com/jsa-aerial/DMM/blob/master/doc/IBM-AI-Systems-Day-2018/aisys18-bukatin.pdf`
Reference paper: `https://arxiv.org/abs/1712.07447`

2005-2015: Research collaboration on **Partial Metrics**.

We discovered deep connections between partial metrics and fuzzy equalities on the sets of partially defined elements in 2006 and studied and understood the nature of those connections culminating in a *Fuzzy Sets and Systems* paper in 2014. We also published a review paper in *American Mathematical Monthly* in 2009.

LIST OF PUBLICATIONS: `https://anhinga.github.io/brandeis-mirror/publications-2024.pdf`

TEACHING: 1992-1996

Boston University, Boston, MA. Teaching a graduate course in computer science: Formal Semantics of Programming Languages.
Framingham State College, Framingham, MA. Teaching courses in computer science: Programming Languages (Scheme, Prolog), Data Structures (C++). IBM PC.
Brandeis University, Waltham, MA. Helping to teach courses in computer science: Structure and Interpretation of Programming Languages (Scheme), Theory of Computation, Mathematical Logic in Computer Science (graduate), Semantics of Programming Languages and Lambda-calculus (with ML - graduate). IBM PC, UNIX workstations.

EDUCATION:

09/1992-02/2002: Brandeis University.

PhD in Computer Science.

Semantics of Programming Languages: its Mathematics and Applications.

My graduate research focused on mathematics of *domains for denotational semantics* and its possible applications. Elements of such domains were used to express meanings of programs. In 1996-1998 we obtained new striking results in the theory of continuous generalized metrics (partial and relaxed metrics) and measures on domains. In particular, they enabled us to compute *meaningful distances between programs*. Our earlier results advanced the approach representing meanings of programs as theories in logical calculi. In particular, an original notion of *subtyping* was developed.

4 publications in *Lecture Notes in Computer Science* (Springer-Verlag) and *Topology Proceedings*.
My papers, manuscripts, and dissertation in computer science are available at
`https://www.cs.brandeis.edu/~bukatin/papers.html`

09/1981-06/1986: Moscow Institute of Railway Engineering.

MS in Applied Mathematics. Thesis on *syntax-oriented program editors* (Pascal, IBM PC).

PERSONAL: US Citizen.