

# Trevor A. Brown

**Postdoctoral researcher at Institute of Science and Technology, Austria**

Computer Science Department      Phone: +1 (647) 286-0462  
Am Campus 1                              E-mail: me [at] tbrown.pro  
A – 3400 Klosterneuburg                Web: <http://tbrown.pro>

## Education

**University of Toronto**, Toronto, Canada, Jan. 2013 – Jun. 2017

Ph.D. Computer Science (supervised by Faith Ellen); A+ GPA

Thesis: Techniques for constructing efficient lock-free data structures (312 pages)

**University of Toronto**, Toronto, Canada, Sep. 2011 – Jan. 2013

M.Sc. Computer Science (supervised by Faith Ellen); A+ GPA

Thesis: Pragmatic primitives for non-blocking data structures

**York University, First Class with Distinction**, Toronto, Canada, 2006 – 2011

B.Sc. Hons. Major Computer Science, Hons. Minor Mathematics; A+ GPA

## Research Interests

- Concurrent data structures
- Transactional memory
- Non-blocking algorithms
- Memory management
- Non-volatile memory
- Databases

## Employment

**Postdoctoral Researcher.** IST Austria (Oct 2017 – present).

Working with Professor Dan Alistarh.

**Postdoctoral Researcher.** Technion, Israel Institute of Technology (Mar – Oct 2017).

Worked with Professor Hagit Attiya.

**Research Intern.** Oracle Labs East, Scalable Synchronization Group (Summer 2015).

Designed and implemented adaptive transactional lock-elision algorithms and work delegation algorithms for large scale systems with non-uniform memory architectures.

**Application Developer.** UPS Canada (2005 – 2006, 2008 – 2009).

Redesigned internal software to scale under a hugely increased user load.

Contributed over 200,000 lines of production code.

## Conference and Journal Publications

1. **Getting to the root of concurrent search tree performance.**  
Maya Arbel-Raviv, Trevor Brown and Adam Morrison.  
Submitted.
2. **Harnessing epoch-based reclamation for efficient range queries.**  
Maya Arbel-Raviv and Trevor Brown.  
23<sup>rd</sup> ACM Symp. on Princ. and Practice of Parallel Programming (**PPoPP'18**), to appear.
3. **Reuse, don't recycle: transforming lock-free algorithms that throw away descriptors.**  
Maya Arbel-Raviv and Trevor Brown. [[Paper](#)] [[Slides](#)] [[Video](#)]  
31<sup>st</sup> ACM Symposium on Distributed Computing (**DISC'17**), pp 4:(1-16).
4. **Cost of concurrency in hybrid transactional memory.**  
Trevor Brown and Srivatsan Ravi. [[Paper](#)] [[Slides](#)]  
31<sup>st</sup> ACM Symposium on Distributed Computing (**DISC'17**), pp 9:(1-16).
5. **A template for implementing fast lock-free trees using HTM.**  
Trevor Brown. [[Paper](#)] [[Slides](#)]  
36<sup>th</sup> ACM Symp. on the Principles of Distributed Computing (**PODC'17**), pp 293-302.
6. **PHyTM: persistent hybrid transactional memory.**  
Hillel Avni and Trevor Brown.  
Very Large Data Bases Journal Volume 10(4), 12 pages.  
*Full version of the paper at VLDB'17.*
7. **PHyTM: persistent hybrid transactional memory.**  
Hillel Avni and Trevor Brown. [[Paper](#)] [[Slides](#)]  
43<sup>rd</sup> International Conference on Very Large Data Bases (**VLDB'17**), pp 409-420.
8. **Investigating the performance of hardware transactions on a multi-socket machine.**  
Trevor Brown, Alex Kogan, Yossi Lev and Victor Luchangco. [[Paper](#)] [[Slides](#)]  
28<sup>th</sup> ACM Symp. on Parallelism in Algorithms and Architectures (**SPAA'16**), pp 121-132.
9. **Reclaiming memory for lock-free data structures: there has to be a better way.**  
Trevor Brown. [[Paper](#)] [[Slides](#)]  
34<sup>th</sup> ACM Symp. on the Principles of Distributed Computing (**PODC'15**), pp 261-270.
10. **B-slack trees: space efficient B-trees.**  
Trevor Brown. [[Paper](#)] [[Slides](#)]  
14<sup>th</sup> Scandinavian Symp. and Workshops on Algorithm Theory (**SWAT'14**), pp 122-133.
11. **A general technique for non-blocking trees.**  
Trevor Brown, Faith Ellen and Eric Ruppert. [[Paper](#)] [[Slides](#)]  
19<sup>th</sup> ACM Symp. on Princ. and Practice of Parallel Programming (**PPoPP'14**), pp 329-342.
12. **Pragmatic primitives for non-blocking data structures.**  
Trevor Brown, Faith Ellen and Eric Ruppert. [[Paper](#)] [[Slides](#)]  
31<sup>st</sup> ACM Symposium on the Principles of Distributed Computing (**PODC'13**), pp 13-22.

**13. Range queries in non-blocking k-ary search trees.**

Trevor Brown and Hillel Avni. [[Paper](#)]

16<sup>th</sup> International Conf. on Principles of Distributed Systems (**OPODIS'12**), pp 31-45.

**14. Non-blocking k-ary search trees.**

Trevor Brown and Joanna Helga. [[Paper](#)] [[Slides](#)] [[Video](#)]

15<sup>th</sup> International Conf. on Principles of Distributed Systems (**OPODIS'11**), pp 207-221.

## Workshop Papers

**15. Cost of concurrency in hybrid transactional memory.**

Trevor Brown and Srivatsan Ravi. [[Paper](#)] [[Slides](#)]

12<sup>th</sup> ACM SIGPLAN Workshop on Transactional Computing (**TRANSACT'17**), 8 pages.  
*Preliminary version of the conference paper at DISC'17.*

**16. Persistent hybrid transactional memory.**

Hillel Avni and Trevor Brown. [[Paper](#)] [[Slides](#)]

11<sup>th</sup> ACM SIGPLAN Workshop on Transactional Computing (**TRANSACT'16**), 8 pages.  
*Preliminary version of the conference paper at VLDB'17.*

**17. Investigating the performance of hardware transactions on a multi-socket machine.**

Trevor Brown, Alex Kogan, Yossi Lev and Victor Luchangco. [[Paper](#)] [[Slides](#)]

11<sup>th</sup> ACM SIGPLAN Workshop on Transactional Computing (**TRANSACT'16**), 8 pages.  
*Preliminary version of the conference paper at SPAA'16.*

## Short Papers and Posters

**18. Reuse, don't recycle: transforming lock-free algorithms that throw away descriptors.**

Maya Arbel-Raviv and Trevor Brown. [[Paper](#)] [[Poster](#)]

22<sup>nd</sup> ACM Symp. on Princ. and Practice of Parallel Programming (**PPoPP'17**), pp 429-430.  
*Poster and short version of conference paper at DISC'17.*

**19. Concurrent data structures.**

Faith Ellen and Trevor Brown. [[Paper](#)] [[Slides](#)]

35<sup>th</sup> ACM Symp. on the Principles of Distributed Computing (**PODC'16**), pp 151-153.  
*Short paper to accompany an invited talk by Faith Ellen at PODC'16.*

**20. Faster data structures in transactional memory using three paths.**

Trevor Brown. [[Paper](#)] [[Slides](#)]

29<sup>th</sup> ACM Symposium on Distributed Computing (**DISC'15**), pp 671-672.  
*Short version of conference paper at PODC'17.*

**21. A general technique for non-blocking trees.**

Trevor Brown, Faith Ellen and Eric Ruppert. [[Paper](#)] [[Slides](#)]

27<sup>th</sup> ACM Symposium on Distributed Computing (**DISC'13**), pp 567-568.  
*Short version of conference paper at PPoPP'14.*

## Articles and Technical Reports

22. **Reuse, don't recycle: transforming lock-free algorithms that throw away descriptors.**  
Maya Arbel-Raviv and Trevor Brown. [[Paper](#)]  
ArXiv Computing Research Repository (CoRR), abs/1708.01797, 32 pages.  
*Full version of conference paper at DISC'17.*
23. **A template for implementing fast lock-free trees using HTM.**  
Trevor Brown. [[Paper](#)]  
ArXiv Computing Research Repository (CoRR), abs/1708.04838, 20 pages.  
*Full version of conference paper at PODC'17.*
24. **Techniques for constructing efficient lock-free data structures.**  
PhD thesis, University of Toronto. [[Paper](#)]  
Committee: Faith Ellen (University of Toronto), Azadeh Farzan (University of Toronto), Vassos Hadzilacos (University of Toronto), Maurice Herlihy (Brown University), Ryan Johnson (University of Toronto), Sam Toueg (University of Toronto).  
ArXiv Computing Research Repository (CoRR), abs/1712.05406, 312 pages.
25. **Reclaiming memory for lock-free data structures: there has to be a better way.**  
Trevor Brown. [[Paper](#)]  
ArXiv Computing Research Repository (CoRR), abs/1712.01044, 27 pages.  
*Full version of conference paper at PODC'15.*
26. **B-slack trees: space efficient B-trees.**  
Trevor Brown. [[Paper](#)]  
ArXiv Computing Research Repository (CoRR), abs/1712.05020, 19 pages.  
*Full version of conference paper at SWAT'14.*
27. **A general technique for non-blocking trees.**  
Trevor Brown, Faith Ellen and Eric Ruppert. [[Paper](#)]  
ArXiv Computing Research Repository (CoRR), submitted, 41 pages.  
*Full version of conference paper at PPOPP'14.*
28. **Pragmatic primitives for non-blocking data structures.**  
Trevor Brown, Faith Ellen and Eric Ruppert. [[Paper](#)]  
ArXiv Computing Research Repository (CoRR), submitted, 47 pages.  
*Full version of conference paper at PODC'13.*
29. **Range queries in non-blocking k-ary search trees.**  
Trevor Brown and Hillel Avni. [[Paper](#)]  
ArXiv Computing Research Repository (CoRR), abs/1712.05101, 17 pages.  
*Full version of conference paper at OPODIS'12.*
30. **Non-blocking k-ary search trees.**  
Trevor Brown and Joanna Helga. [[Paper](#)]  
York University Technical Report CSE-2011-04, 52 pages.  
*Full version of conference paper at OPODIS'11.*

## Patent Applications

31. **Adaptive techniques for improving performance of hardware transactions on multi-socket machines.** Oracle Labs, US patent application #20170075720, 34 pages. Alex Kogan, Yossi Lev, Victor Luchangco and Trevor Brown. [[Link](#)]

## Conference Talks and Invited Presentations

### **Good data structure experiments are R.A.R.E.**

- Invited talk at the 1<sup>st</sup> Workshop on the Theory and Practice of Concurrency, Vienna, Austria (Oct 2017). *Held in conjunction with DISC'17*. [[Slides](#)] [[Video](#)]
- Oath/Yahoo! Labs, Haifa, Israel (Sep 2017). [[Slides](#)]

### **Reuse, don't recycle: transforming lock-free algorithms that throw away descriptors.**

- 31<sup>st</sup> ACM Symposium on Distributed Computing, Vienna, Austria (Oct 2017). [[Slides](#)] [[Video](#)]

### **Cost of concurrency in hybrid transactional memory.**

- 31<sup>st</sup> ACM Symposium on Distributed Computing, Vienna, Austria (Oct 2017). [[Slides](#)]

### **A template for implementing fast lock-free trees using HTM.**

- 36<sup>th</sup> ACM Symposium on the Principles of Distributed Computing, Washington, United States (Jul 2017). [[Slides](#)]

### **Techniques for Constructing Efficient Lock-free Data Structures.**

- Technion - Israel Institute of Technology, Haifa, Israel (May 2017). [[Slides](#)]
- University of Toronto, Toronto, Canada (Mar 2017). [[Slides](#)]

### **Investigating the performance of hardware transactions on a multi-socket machine.**

- 28<sup>th</sup> ACM Symposium on Parallelism in Algorithms and Architectures, Monterey, United States (Jul 2016). [[Slides](#)]
- 11<sup>th</sup> ACM SIGPLAN Workshop on Transactional Computing, Barcelona, Spain (Mar 2016). [[Slides](#)]

### **Persistent hybrid transactional memory.**

- 11<sup>th</sup> ACM SIGPLAN Workshop on Transactional Computing, Barcelona, Spain (Mar 2016). [[Slides](#)]

### **Faster data structures in transactional memory using three paths.**

- 29<sup>th</sup> ACM Symposium on Distributed Computing, Tokyo, Japan (Oct 2015). [[Slides](#)]

### **Scalable transactions on NUMA systems.**

- Oracle Labs East, Burlington, United States (Aug 2015). [[Slides](#)]

### **Reclaiming memory for lock-free data structures: there has to be a better way.**

- 34<sup>th</sup> ACM Symposium on the Principles of Distributed Computing, San Sebastian, Spain (Jul 2015). [[Slides](#)]

### **Hardware transactional memory and the lemming effect.**

- University of Toronto, Toronto, Canada (Apr 2015). [[Slides](#)]

**Java Experiments on MTL: From past mistakes to best practices.**

- York University, Toronto, Canada (Mar 2015). [[Slides](#)]

**Memory reclamation for lock-free data structures.**

- University of Toronto, Toronto, Canada (Aug 2014).

**B-slack trees: space efficient B-trees.**

- 14<sup>th</sup> Scandinavian Symposium and Workshops on Algorithm Theory, Copenhagen, Denmark (Jun 2014). [[Slides](#)]

**A general technique for non-blocking trees.**

- 19<sup>th</sup> ACM Symposium on Principles and Practice of Parallel Programming, Orlando, United States (Feb 2014). [[Slides](#)]

- 27<sup>th</sup> ACM Symposium on Distributed Computing, Jerusalem, Israel (Aug 2013). [[Slides](#)]

**Pragmatic primitives for non-blocking data structures.**

- 31<sup>st</sup> ACM Symposium on the Principles of Distributed Computing, Montreal, Canada (Jul 2013). [[Slides](#)]

- University of Toronto, Toronto, Canada (Jul 2013).

**Building a non-blocking chromatic tree.**

- TransForm School on Research Directions in Distributed Computing, Heraklion, Crete (Jun 2013). [[Slides](#)]

**Range queries in non-blocking k-ary search trees.**

- 16<sup>th</sup> International Conference on Principles of Distributed Systems, Rome, Italy (Nov 2012).

**Non-blocking k-ary search trees.**

- 15<sup>th</sup> International Conference on Principles of Distributed Systems, Toulouse, France (Oct 2011). [[Slides](#)] [[Video](#)]

**Experiences with Intel's Multicore Testing Lab.**

- York University, Toronto, Canada (2011).

## Teaching Experience

- *Teaching Assistant – University of Toronto* (2011 – 2014)
  - CSC2221: graduate theory of distributed computing (2013).
  - CSC263: data structures and analysis (2011, 2012, 2013, 2014).
  - CSC369: principles of operating systems (2012).
  - CSC265: enriched data structures and analysis (2011).
- *Programming contest coach – York University* (2009)
  - Held a seminar series on algorithms and problem solving. Gave short lectures, involving students in the process of solving algorithmic problems, then directed hands-on problem solving sessions, and held practice programming contests.

## Supervisory Experience

(undergraduate research projects)

- Jialin Song: designing a locking scheme that bridges the gap between fine-grained and coarse-grained locking (University of Toronto, 2014).
- Ken Hoover: designing and implementing a non-blocking relaxed AVL tree, and a non-blocking relaxed (a,b)-tree (University of Toronto, 2013).

## Software Artifacts

(<http://implementations.tbrown.pro>)

- **Java lock-free data structure library (22,000 lines of code)**  
Produced the first lock-free unbalanced binary search tree implementation.  
Produced the first implementation of LLX and SCX synchronization primitives.  
Also introduced: k-ary search trees, relaxed AVL trees, Chromatic trees, b-slack trees.  
Includes experimental test harness with support for 13 competing data structures.
- **C++ lock-free data structure library (23,000 lines of code)**  
Produced the first C/C++ implementation of LLX and SCX synchronization primitives.  
Implemented unbalanced BSTs and relaxed (a,b)-trees using LLX and SCX.  
Produced four different transactional memory based algorithms for each data structure.
- **Lock-free memory reclamation in C++ (9,000 lines of code)**  
Provided a record manager library with allocation, reclamation and object pooling plugins (including five allocators and four memory reclamation algorithms).  
Implemented lock-free BSTs and Chromatic trees using this library.
- **Reusable descriptors for lock-free data structures in C++ (29,000 lines of code)**  
Provided a lock-free reusable descriptor library.  
Used this library to accelerate four advanced lock-free data structures.
- **Support for range query operations in C++ (55,000 lines of code)**  
Produced three novel algorithms for adding range query operations to data structures.  
Augmented seven different data structures with range query support (producing up to five variants of each data structure).  
Used these data structures to accelerate an in-memory database (DBx1000).
- **PHyTM: non-volatile hybrid transactional memory for databases (proprietary code – Huawei Research)**  
Implemented hybrid TM on a simulated system with non-volatile DRAM.  
Accelerated two in-memory databases by using PHyTM to synchronize threads (with large improvements over 2-phase locking and optimistic concurrency control).
- **Hardware lock elision for multsocket systems (proprietary code – Oracle Labs)**  
Created a drop-in library for accelerating arbitrary lock-based code on systems with non-uniform memory architectures and hardware transactional memory.  
Demonstrated large speedups for data structures, benchmarks, and applications.

## Grants and Fellowships

<b>NSERC</b> Postdoctoral fellowship (\$90,000) Second highest ranked applicant	2017 – 2018
<b>NSERC</b> Alexander Graham Bell CGS D3 (\$105,000)	2013 – 2016
<b>University of Toronto</b> PhD fellowship (\$31,000)	2013 – 2016
<b>University of Toronto</b> MSc fellowship (\$19,000)	2011 – 2012
<b>NSERC</b> Alexander Graham Bell CGS M (\$17,500)	2011
<b>NSERC</b> Undergraduate Research Awards (\$22,000)	2009, 2010, 2011
<b>York</b> University Entrance/Continuing Scholarships (\$3,720)	2006 – 2010
<b>York</b> Professor Ruth Hill Memorial Award (\$1,045) Awarded to the top undergraduate student in the Faculty of Science and Engineering	2008

## Awards

- Award for Excellence in Teaching Assistance (U of T CS Student Union, 2014).
- Faculty of Science and Engineering Silver Medal (York University, 2011).
- Honourable Mention in the Computing Research Association's Outstanding Undergraduate Researcher Award competition (2011).
- Represented York at the ACM International Collegiate Programming Contests, ECNA region (2007 – 2011). In 2009, led senior school team to finish 6<sup>th</sup> (of 115).
- Dean's Honour Roll (York University) – Science and Engineering (2007 – 2011).
- Chair's Honour Roll (York University) – Mathematics and Statistics (2010).

## Community Service

- **Program committee member** for ICDCS'18.
- **Conference paper reviews** for PODC'11, DISC'12, PODC'13, DISC'14, PODC'14, PPOPP'16, SIROCCO'16, SPAA'16, DISC'16, SPAA'17, RANDOM'17, DISC'17.
- **Artifact evaluator** for PPOPP'16.
- **Journal reviews** for the ACM Journal of Distributed Computing (DC/DIST), the IEEE Journal of Transactions on Parallel and Distributed Systems (TPDS), and the Elsevier Journal of Logical and Algebraic Methods in Programming (JLAMP).
- **Search committee for Chair of Computer Science**, University of Toronto (2015)
- **Wrote tenure recommendation letters**, York University
- **Developed student study resources** (2009 – 2010)  
Developed online student resources for courses in Operations Research and Japanese, and for the ACM ICPC programming contests. They have been accessed 250,000 times.
- **Volunteer note-taker** (2008 – 2009)  
Tutored and took notes on behalf of students with accessibility needs.