Reversible Algorithms

JAYSON LYNCH
JAYSON.LYNCH @ UWATERLOO.CA
UNIVERSITY OF WATERLOO

Topics in Reversible Algorithms

- Software and Programming Languages [Glück et al ‘20]
  - Compilers and Interpreters [Yokoyama, Glück ‘07] [Yokoyama, Axelson, Glück ‘08]
  - Memory management [Cservenka, Haulund, Mogensen, Glück ‘18]
  - Object oriented [Axelson, Shultz ‘16] [Haulund, Mogensen, Glück ‘17] and functional programming [Yokoyama, Axelson, Glück ‘11] [Kawabe, Glück ‘03]
- Algorithms and Complexity Theory
  - Universal transformations for Turing Machines and Circuit Models [later slides]
  - Pebbling lower bounds [Li, Vitányi ‘92] [Li, Vitányi ‘96]
  - Oracle separation [Frank, Ammer ‘17]
  - Classification of reversible circuits [Aaronson, Grier, Schaeffer ‘15]
- Efficient Reversible Algorithms [later slides]
  - Small constant factor overhead in time and space complexity
  - Sorting Algorithms
  - Graph Algorithms
  - Linear Algebra

Universal Reversible Computing

- History Recording [Lecerf '63, Bennett '73]
  - Make functions bijective by storing inputs
  - Time: $T'(n)=O(T(n))$; Space: $S'(n)=O(S(n)\,T(n))$
- Recursive Recomputing [Bennett '79]
  - Compute to the midpoint, store it and uncompute to the last checkpoint. Recurse.
  - Time: $T'(n)=O(T(n)\lg(T(n)))$; Space: $S'(n)=O(S(n)\lg(T(n)))$
- Configuration Space Enumeration [Lange, McKenzie, Tapp '00]
  - Walk the entire computation tree. Reminiscent of Savitch's Algorithm.
  - Time: $T'(n)=O(2^{T(n)})$; Space: $S'(n)=O(S(n))$
- Time-Space Tradeoff [Williams '00][Buhrman, Tromp, Vitanyi '01]
  - Embed Configuration Space Enumeration at the bottom of a Bennett recursion.
  - Time: $T'(n)=O(S(n)k2^{T(n)/2^k})$; Space: $S'(n)=O(kS(n))$
- Computing with Dirty Ancilla Bits [Xu '15]
  - Space can be temporarily used without knowing it's prior state.
  - Time: $T'(n)=O(2^{T(n)})$; Space: $S'(n)=S(n)+1$

Energy, Entropy, and Conditional Reversibility

- Tradeoff between space-usage and bit-erasure as a resource [Li, Vitányi '92]
- Define word RAM and transdichotomous RAM models with energy cost based on function injectivity [Demaine, Lynch, Mirano, Tyagi '16]
- Formally defines conditional reversibility, allowing reversibility on input-restricted domains [Frank '17]
- Analysis of conditional reversibility in a machine learning test case. [DeBenedictis, Frank, Anderson '16]

What is Efficient?

- Asymptotically equivalent time and space usage
- Small constant factors in overhead
- Axelsen and Yokoyama define the following:
  - $g(n)$ *faithful* simulation has time $T'(n)=\Theta(T(n))$ and space $S'(n)=O(S(n)+g(n))$
  - A *hygienic* simulation is $g(n)$ faithful for minimum possible $g(n)$.

Efficient Reversible Algorithms

- Sorting Algorithms [Axelsen, Yokoyama '15] [Masuda, Yokoyama '19]
  - Constant factor overhead for several algorithms
  - Quadratic algorithms which preserves best case running time with additive space overhead
- Graph Algorithms [Frank '99] [Nøhr '15] [Guo, Peng, He '18]
  - Shortest Path and APSP
  - Minimum Spanning Tree
- Data Structures [Yokoyama, Axelsen, Glück '08] [Axelsen, Glück '13] [Nøhr '15] [Demaine, Lynch, Mirano, Tyagi '16]
  - Adjacency List
  - Binary Search Tree
  - Dynamic Array
  - Disjoint Set
  - Min-priority Queue
- FFT [Yokoyama, Axelsen, Glück '08]
- Matrix Multiply [Frank '99] [Demaine, Lynch, Mirano, Tyagi '16]

Techniques for Efficient Reversible Algorithms

- Input logging
- Reversible sub-routines
  - Input can be calculated from the output
  - Call and Uncall the routine, saving space when outside the function
- Paired branches and protected conditionals
  - Control flow must know where to return
  - Conditionals unedited inside a loop are easier to maintain
- Pointer swapping
  - Always maintain back pointers
  - Instead of destroying a pointer, change where it is stored
- Permutation representations
  - Permutations are bijective

Special Purpose Reversible Computing

- Accelerators and heterogeneous architectures
  - Hardware designed to be extremely performant on certain classes of algorithms.
  - GPU, TPU, ASIC, etc.
  - Restricted computing classes could make reversible architecture design easier.
- Embedded/ubiquitous computing and extreme environments
  - Often requires extreme energy efficiency.
  - Often only needs special purpose computation.
- High-performance computing
  - Massively parallelizable algorithms, again a restricted class.
  - Willing to invest in hardware to get performance.
- Quantum computing
  - Requires substantial classical computation to control the quantum computation.
  - The classical computing must interface with quantum circuits.
  - Already uses adiabatic hardware, ex. superconducting circuits.

Special Purpose Reversible Computing

- Special classes of algorithms
- Requires low-energy / high-performance computing
- Higher investment costs acceptable

- Requires study and optimization of specific classes of algorithms
- Software/Hardware co-design
- Provides targets and stepping stones for both algorithmic and hardware advances

Suggested Targets

- Optimization algorithms
- Machine Learning
- Differential equation solvers
- Computational Geometry
  - Triangulation
  - Point/range query
  - Ray/polygon intersection
- Perfect matchings
- Edit distance and other string comparison

Bibliography

- Scott Aaronson, Daniel Grier, and Luke Schaeffer. The classification of reversible bit operations. CoRR, abs/1504.05155, 2015. URL: http://arxiv.org/abs/1504.05155.
- Charles H. Bennett. Logical reversibility of computation. IBM Journal of Research and Development, 17(6):525–532, 1973
- Charles H. Bennett. Time/space trade-offs for reversible computation. SIAM Journal on Computing, 18(4):766–776, August 1989.
- Harry Buhrman, John Tromp, and Paul Vit´anyi. Time and space bounds for reversible simulation. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, Proceedings of the 28th International Colloquium on Automata, Languages and Programming, volume 2076 of Lecture Notes in Computer Science, pages 1017–1027. Springer Berlin Heidelberg, 2001. URL: http://dx.doi.org/10.1007/3-540-48224-5 82, doi:10.1007/3-540-48224-5_82.
- DeBenedictis, E. P., Frank, M. P., Ganesh, N., & Anderson, N. G. (2016, October). A path toward ultra-low-energy computing. In *2016 IEEE International Conference on Rebooting Computing (ICRC)* (pp. 1-8). IEEE.
- Erik D Demaine, Jayson Lynch, Geronimo J Mirano, and Nirvan Tyagi. Energy-efficient algorithms. In Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, pages 321–332, 2016.
- Frank, Michael P. "Foundations of generalized reversible computing." *International Conference on Reversible Computation*. Springer, Cham, 2017.
- Michael P Frank and M Josephine Ammer. Relativized separation of reversible and irreversible space-time complexity classes. arXiv preprint arXiv: 1708.08480, 2017
- Michael Patrick Frank. Reversibility for efficient computing. PhD thesis, Massachusetts Institute of Technology, 1999.
- Lanying Guo, Chao Peng, and Cheng He. New reversible computing algorithms for shortest paths problem. In Proceedings of the 6th International Conference on Information Technology: IoT and Smart City, pages 82–87, 2018.
- Ming Li and Paul Vit´anyi. Reversibility and adiabatic computation: trading time and space for energy. Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 452(1947):769–789, 1996.
- Hiroki Masuda and Tetsuo Yokoyama. Analyzing trade-offs in reversible linear and binary search algorithms. arXiv preprint arXiv:1910.10406, 2019.
- VN Sarah. Reversible graph algorithms. PhD thesis, Master Thesis, University of Copenhagen, 2015.
- Ryan Williams. Space-efficient reversible simulations. Technical report, DIMACS REU report, 2000. http://web.stanford.edu/~rrwill/spacesim9 22.pdf.
- Siyao Xu. Reversible logic synthesis with minimal usage of ancilla bits. arXiv preprint arXiv:1506.03777, 2015.