Structure Of Generating Sets For Reversible Computations

JAAK HENNO
Tallinn University of Technology, Estonia
jaak@cc.ttu.ee

IBM researcher Rolf Landauer stated in 1961 law [1], which connects computations with physics (thermodynamics): when a computational system erases a bit of information, it must dissipate $\ln(2)*kT$ energy (heat); here k is the Boltzmann's constant and T is the temperature; currently this law is referred as the Landauer's principle, but the idea of equivalence of information and thermodynamic entropy was considered already e.g. by Szilard [2]..

In order not to dissipate heat computation should not erase anything. Such a computation is reversible - every step of the computation can be done also in backwards (undo). A reversible computation does not generate heat and according to current knowledge it can be implemented on quantum level, using qubits instead of ordinary bits. But reversible computations/functions occur also in ordinary computations - cryptographic functions, many image-editing functions (lossless image compression) etc all should be reversible. A quantum computer could execute many currently difficult computational tasks much quicker, e.g. a quantum algorithm can solve the integer factorization problem exponentially faster using than the best-known classical algorithms [3]. Thus in recent years reversible computations has become a very important research topic for its enormous possibilities in low power CMOS design, quantum computing and nanotechnology.

The binary Boolean functions - $\&, \vee, \rightarrow, \leftrightarrow, \oplus, \mathrm{NAND}$ etc are not reversible, but negation is.

However, every computation can be embedded into a reversible one - every Turing machine can be made reversible [4]. Among several constructs for converting non-reversible functions into reversible ones most often is cited the Toffoli construction [5] ('Toffoli gate') - a ternary Boolean function with ternary output (a reversible function should have the same number of inputs-outputs), where result of conjunction $x_1 \& x_2$ of the first two arguments is used to flip the state of the third argument, i.e. it implements the 3-ary reversible function

$$(x_1, x_2, x_3) \Rightarrow (x_1, x_2, (x_1 \& x_2) \oplus x_3)$$

Usually reversible computations are considered only for binary, i.e. Boolean logic, but e.g. life encodes its programs (genes) in a 4-valued logic.

For 'real' computing one of the most essential problems are bases - sets of functions used to express (calculate) every other function; functions of a base are implemented either in hardware (processor) or using processor functions (compiler). Every n-ary reversible function of m-valued logic implements a substitution on the set of m^n n-place vectors (inputs to outputs) whose coordinates are from the set $\{0,1,...,m-1\}$, i.e. is an element of the group S_{m^n}. Generating sets for the symmetric substitution groups have been studied extensively for quite a time (see e.g. [6],[7]), but for 'real-word' implementations derived from algebraic group-theoretic properties of the group S_{m^n} may not be always what is best/cheapest to implement.

Problems of cost of implementation and usability of bases in processors (implementing non-reversible computations) have been studied in for quite a long time. The Toffoli idea (storing intermediate values fliping the state of some wire) has been used to show that if a Boolean function can be embedded into an even permutation with polynomial-size cycle representation then the function can be implemented by a polynomial-size reversible circuit [8].

Here will be shown that the Toffoli idea can be used to convert every generating set of functions (base) in m-valued logic into generating set (base) of reversible functions of m-valued logic; the idea is similar to what has been used in [9]. This construction preserves partial order of bases [10], based on (minimal) depth $d_{\mathcal{F}}(f)$ of implementation of function f in base $\mathcal{F}$: $\mathcal{F}_1 \leq_d \mathcal{F}_2$ iff there exists a constant k such that for every function f of m-valued logic $\mathrm{d}_{\mathcal{F}_1}(f) \leq \mathrm{d}_{\mathcal{F}_2}(f) + k$; for binary Boolean bases $k = 2$.

References

[1] R.Landauer, 'Information is Inevitably Physical', in 'Feynman and Computation' ed. A. J.G.Hey (Addison Wesley Longman, Reading MA 1998)

[2] Szilard, L., 1929, "On the Decrease of Entropy in a Thermodynamic System by the Intervention of Intelligent Beings", Zeitschrift fur Physik 53: 840–856. English translation in The Collected Works of Leo Szilard: Scientific Papers, B.T. Feld and G. Weiss Szilard (eds.), Cambridge, Massachusetts: MIT Press, 1972, pp. 103–129.

[3] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing, 26(5):1484-1509, 2005.

[4] C. H. Bennett, "Logical reversibility of computation," IBM Journal of Research and Development, vol. 17, no. 6, pp. 525-532, 1973.

[5] T. Toffoli, "Reversible Computing," Tech. Memo MIT/LCS/TM-151, MIT Lab for CS, '80.

[6] G.A. Miller. Theory and applications of finite groups. HF Blichfeldt, IE DICKSON - 1916; URL: http://www.ams.org/journals/tran/1928-030-01/S0002-9947-1928-1501419-9/S0002-9947-1928-1501419-9.pdf

[7] G.A. Miller. Possible orders of two generators of the alternating and of the symmetric group. Bulletin of the American Mathematical Society, vol. 7 (1901), p.424

[8] A. Brodsky. "Reversible Circuit Realizations of Boolean Functions", Proceedings of the 3rd IFIP International Conference on Theoretical Computer Science, 2004.

[9] De Vos A., Desoete B., Janiak F., Nogawski A. Control gates for reversible computers. Proc. 11th Int. Workshop on Power and Timing Modeling, Optimization and Simulation, Yverdon, Sept. 2001, 9.2.1–9.2.10

[10] J. Henno. On equivalent sets of functions. Discrete Applied Mathematics, Vol 4, 2, 1982, pp 153–156