Generic Quantum Block Compression

John Langford *

April 12, 2002

Abstract

A generic approach for compiling any classical block compression algorithm into a quantum block compression algorithm is presented. Using this technique, compression asymptoticaly approaching the von Neumann entropy of a qubit source can be achieved. The automatically compiled algorithms are competitive (in time and space complexity) with hand constructed quantum block compression algorithms.

1 Context

We are concerned with quantum compression, which is best understood in the context of quantum information theory. Informally, the Schumacher coding theorem [1] [2] states that a qubit source can be compressed to its von Neumann entropy with high fidelity. This statement is strongly analogous to the Shannon coding theorem for classical information theory which states that a bit source can be compressed to its Shannon entropy with high probability. Quantum compression is therefore concerned with algorithms for achieving compression near to the von Neumann entropy of a qubit source in a reasonable amount of time and space.

*This work was also partially completed at ATnT Shannon Labs. Address: Computer Science Department, CMU, 5000 Forbes Avenue, Pittsburgh, PA 15213

The main result of this paper is a compilation algorithm which takes as input any family of classical block compression (and decompression) algorithm as well as a description of the quantum source. The output is a quantum block compression and decompresion algorithm with the following properties:

1. If the classical compression algorithm asymptotically approaches Shannon entropy, then the quantum compression algorithm asymptotically approaches the von Neumann entropy.

2. If the classical compression and decompresion algorithm each take space S and time T, then the quantum compression algorithm uses space $O(S \log T)$ and time $O(\frac{1}{\epsilon} ST^{1+\epsilon})$ (for any ϵ).

Several hand constructed algorithms for block quantum data compression have been proposed. These include quantum arithmetic compression [3] which runs in $O(n^3)$ time and a combinatoric index compression algorithm [4] running in $O(n^3)$ time and $O(\sqrt{n})$ space. In further work, the combinatoric index compression was parallelized to a circuit of only polylogarithmic depth [5].

The remainder of this paper will discuss some information theory preliminaries, present the compiler, compare the performance of this compiler with hand crafted approaches, and then close with discussion.

2 (Quantum) Information Theory

Suppose that we have a qubit source producing qubits in the state $|a_i\rangle$ with probability p_i. Then the density matrix of this source will be given by the average outer product:

$$\rho = \sum_i p_i |a_i\rangle\langle a_i| = \sum_i q_i |\phi_i\rangle\langle\phi_i|$$

where $|\phi_i\rangle$ are the eigenvectors of the density matrix and q_i is the eigenvalue of $|\phi_i\rangle$.

We wish to compress the tensor product of n qubits each drawn independently from this source so as to achieve a high "fidelity". Fidelity is defined as the probability that a measurement can distinguish between the input and output states. Let $|b_r\rangle$ be the input state chosen with random bits r, and $|b_r^{'}\rangle$ be the output of a compression/decompression cycle. Then, the fidelity is defined by:

$$F = E_r \langle b_r | b_r^{'} \rangle$$

where E_r is an expectation with respect to the quantum source as well as with respect to the outcome of the quantum measurement.

The Schumacher coding theorem states that the minimum coding rate of any compression scheme with a high expected fidelity is given by the von Neumann entropy:

$$S(\rho) = Tr(\rho \log \rho)$$

Thus, we are particularly motivated by compression schemes which can (asymptotically) achieve the von Neumann entropy.

3 Generic quantum block compression

A generic (classical) block compression algorithm, $C : \{0,1\}^n \to \{0,1\}^m$, maps n bits to m bits with n larger than m. The corresponding decompression algorithm, $D : \{0,1\}^m \to \{0,1\}^n$, will then invert the compression to give the original state, *if the original state falls in the typical subspace.* The typical subspace, as defined by information theory, is the set of "high probability" sequences. A simple counting argument implies that it is not possible to succesfully invert all sequences when $m \leq n$. This argument (in modified form) also applies to quantum information theory.

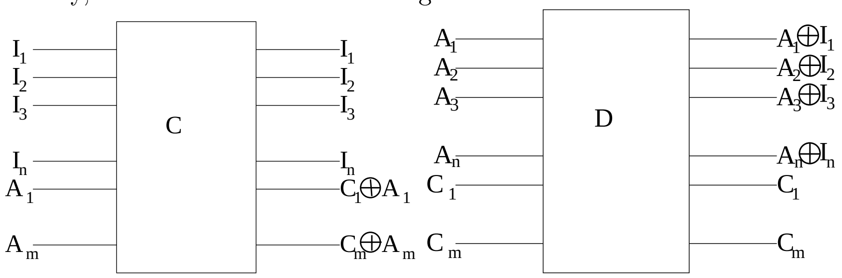
We can use the general result that any classical function can be implemented [6] (reasonably efficiently) on a quantum computer. In particular, If we let $I_1, ..., I_n$ be the bits of the input string and $C_1, ..., C_m$ be the bits of the compressed string then we can compile the compression circuit into the reversible circuit which calculates:

$$C_{\text{rev}}(I_1, ..., I_n, A_1, ..., A_m) = I_1, ..., I_n, A_1 \oplus C_1, ..., A_m \oplus C_m$$

where $A_1, ..., A_m$ are auxiliary bits. The same trick can be done with the decompression circuit to get:

$$D_{\text{rev}}(A_1, ..., A_n, C_1, ..., C_m) = A_1 \oplus I_1, ..., A_n \oplus I_n, C_1, ..., C_m$$

Pictorially, our circuits are the following:



These elements will be our two fundamental building blocks in the construction of generic quantum block compression. Two challenges exist:

1. C_{rev} and D_{rev} both have outputs which are entangled with their input. This is unacceptable in any quantum compression scheme.

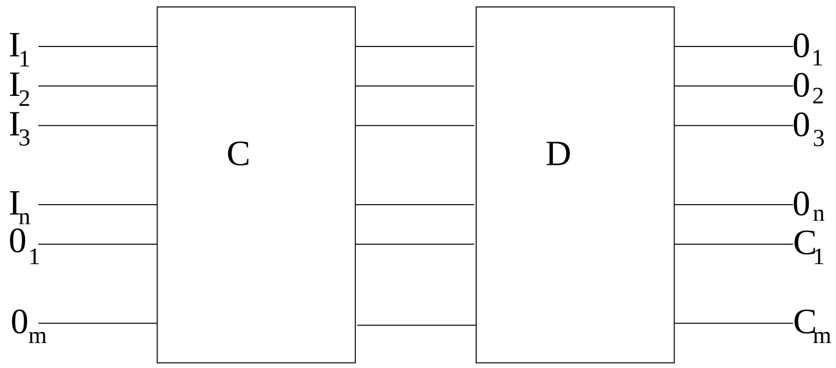
2. There are some quantum sources with a low von Neumann entropy but a large Shannon entropy. We will need to make a circuit which achieves compression near the Shannon entropy to achieve compression near the

von Neumann entropy.

These two challenges are met in the next subsections.

3.1 Entanglement Removal

We are not finished with the compression because the output of C_{rev} leaves the compressed bits entangled with the input bits. There is a simple construction which removes this entanglement: compose the decompressor with the compressor to get the following circuit:



Here we have set the auxiliary bits to 0.

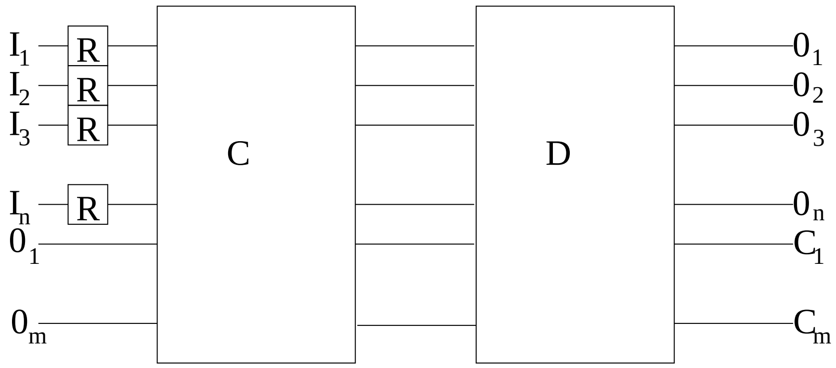
The output of the first compression block is the compressed bits along with the uncompressed bits. The uncompressed bits are then fed into the decompression block instead of 0's, resulting in an output consisting of the compressed bits plus things of the form $I_i \oplus D(C(I))_i$. This term is 0 precisely when the compression is lossless. By assumption, the compression is lossless with high probability implying that we will essentially always measure the 0 vector. At this point, we can measure all of the "0" qubits. If any of our measurements are nonzero, we have detected a low probability event. The process of this measurement will project the wavefunction onto the typical subspace, letting us achieve the projection done explicitly in [3] for "free" as a byproduct of our compression process.

This construction is also particularly elegant because decompression (with entaglement removal) can be accomplished by simply reading the circuit backwards.

In summary, we now have a technique for constructing a quantum compression circuit which outputs the compressed string *without* extra entanglement.

3.2 Achieving von Neumann entropy

The above approach only does *classical* compression of qubits and so it can only achieve the Shannon entropy. In particular, it will fail to compress some highly compressible quantum state distributions such as the state which is $(|0\rangle + |1\rangle)^n$ with probability 1. This state is highly compressible because it has a trivial distribution over possibilities and yet no classical algorithm will compress the qubits well. In order to accomplish compression of this state, we must add an extra rotation to the input and output of the compression and decompression. For example, the operation H^n (Hadamard applied to each bit), would convert the input to $|0^n\rangle$ which is highly compressible. This extra rotation involves $O(n)$ operations since it can be done on each symbol from the quantum source individually giving us the following circuit:



The interpretation of this circuit is simple: we first rotate every qubit to align the eigenbasis and the computational basis. Then, we apply the compression circuit using the decompression circuit to remove entanglement with the input values. Mathematically, we compute

$$D_{\text{rev}}(C_{\text{rev}}(R(I_1), ..., R(I_n), 0_1, .., 0_m))$$

where R is the rotation into the eigenbasis. Once again, decompression is done using the circuit backwards.

This solution works for *any* quantum source. Given a description of any quantum source, the eigenvalue decomposition theorem says that we can construct a matrix R which satisfies:

$$\rho = R \Lambda R^{\top}$$

where Λ is a diagonal matrix of eigenvalues and R is a matrix consisting of eigenvectors. The R matrix will rotate the input source so that the computational basis and the eigenbasis coincide. This rotation does not affect the von Neumann entropy of the source. Now, given that the computational basis and eigenbasis coincide, the von Neumann entropy will be exactly the same as the Shannon entropy. The compiled circuit will therefore achieve the von Neumann limit. To summarize, the compiler is the following:

Make_quantum_compression_circuit(ρ, C, D)

1. Compile the quantum reversible circuits D_{rev} and C_{rev} from the classical circuits D and C.

2. Compose D_{rev} and C_{rev} as in section 3.1.

3. Calculate the eigenvalue matrix, R, given ρ.

4. Rotate all input and output qubits of the quantum circuit by R as in section 3.2.

3.3 Computational efficiency

How efficiently can we compile compression algorithms made for classical (irreversible) computers into reversible quantum mechanical programs? There are a very few results here which are well summarized in [9] which includes work from [7] and [8]. There are a variety of tradeoffs all of which are typically parameterized by the space S and time T of the original irreversible algorithm. One of the more concrete results is that reversible space is $S_{\mathrm{rev}} = S \log T$ and the reversible time is (for any ϵ) $T_{\mathrm{rev}} = \frac{1}{\epsilon} S T^{1+\epsilon}$.

Applying this result to the common "bzip2" block compression algorithm [10] which runs in space $S = O(n)$ and time $T = O(n \log n)$ will require $S_{\mathrm{rev}} \simeq n \log n$ space and $T_{\mathrm{rev}} \simeq \frac{1}{\epsilon} n (n \log n)^{1+\epsilon}$ time (for any ϵ). If we choose $\epsilon < 1$, This is a smaller running time than $O(n^3)$ as used by previous hand crafted quantum block compression algorithms. It is easy to imagine that hand compilation of irreversible compression/decompression algorithms can yield further functional improvements.

4 Conclusion

The technique for automatically compiling classical compression algorithms into quantum compression algorithms gives us a baseline and general structure for comparison with hand crafted quantum compression algorithms. The automatic approach is reasonably efficient and can give results better than current hand crafted approaches in some cases. Hand crafted algorithms which are functionally more efficient than this baseline are a nontrivial improvement.

Schumacher and Westmoreland [11] investigated the use of classical variable length compression in the quantum setting and conclude that quantum compression *must* use a fixed number of (qu)bits. This is a complimentary piece of work which suggests *all* useful quantum compression must be block compression and discusses techniques for converting classical variable length compression into fixed length compression.

The technique of section 3.1 applies to *any* pair of classical algorithms with an inversion property: $A_2(A_1(x)) = x$. This includes encryption and decryption or encoding and decoding. For algorithms with this property, we can construct quantum analogs with a special property: the "inputs" are disentangled from the "outputs". This disentanglement comes at a relatively small price: a factor of 2 increase in the circuit size. The encoder and decoder are *each* constructed from a circuit of size $|A_1| + |A_2|$ rather than size $|A_1|$ and $|A_2|$.

In addition to hand crafting efficient compression algorithms, one other significant problem in quantum data compression exists: removing the requirement for full information about the quantum source. We know this is (information theoretically) possible [12] although no explicit quantum algorithm yet exists that achieves asymptotically efficient compression without knowledge of the source. Intuitively, this is possible because the precision of the specification of the rotation matrix R (in our construction) improves exponentially in the number of (qu)bits required to specify it. The necessary precision required to achieve a constant error rate scales linearly with n, the number of input qubits to be compressed. Consequently, the portion of the bits which must be allocated to specifying R is asymptotically 0. An explicit algorithm for doing universal quantum compression would yield a significant improvement in our understanding of quantum compression.

Acknowledgements

I would like to thank Andris Ambainis, Klejda Bega, David Relyea, Peter Shor, Luis von Ahn, and Ke Yang for advice and comments in preparing this paper.

References

[1] B. Schumacher, "Quantum Coding" Physical Review A, Vol. 51 pp. 2738-2747, 1995.

[2] R. Jozsa and B. Schumacher, "A New Proof of the Quantum Noiseless Coding Theorem", Journal of Modern Optics, 41, 2343 (1994).

[3] I. L. Chuang and D. S. Modha, "Reversible Arithmetic Coding for Quantum Data Compression" IEEE Transactions on Information Theory, Vol. 46, no. 3, pp. 1104-1116.

[4] R. Cleve and D. P. DiVincenzo, "Shumacher's quantum data compression as a quantum computation", Physical Review A, Vol. 54, 4707, 1996.

[5] S. L. Braunstein, C. A. Fuchs, D. Gottesman, and H.-K. Lo, "A quantum analog of Huffman Coding", IEEE Transactions on Information Theory, Vol. 46, No. 4, 1644 (2000).

[6] E. Bernstein and U. Vazirani, "Quantum complexity theory", Special issue on Quantum Computation of the Siam Journal of Computing, Oct. 1997.

[7] C. H. Bennett, "Time/Space Trade-Offs for Reversible Computation", SIAM J. Comput. 18(4): 766 (1989).

[8] R.Y.Levine and A.T. Sherman "A Note on Bennett's Time-Space Tradeoff for Reversible Computation", SIAM J. on Computing 19(4) 673 (1990).

[9] H. Buhrman, J. Tromp, P. Vitanyi, “Time and Space Bounds for Reversible Simulation”, Proc. International Conference on Automata, Languages and Programming, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2001.

[10] J. Seward, The bzip2 block compression algorithm, http://sources.redhat.com/bzip2/

[11] B. Schumacher and M. Westmoreland, quant-ph/0011014

[12] R. Josza, M. Horodecki, P. Horodecki, R Horodecki, “Universal Quantum Information Compression”, Phys. Rev. Letters 81, 1714 (1998).