QUANTUM COMPRESSION

JOHN LANGFORD

Abstract. I describe the motivations for quantum compression, the limits on quantum compression imposed by quantum information theory, and the existing literature on quantum compression. I also develop a general approach for transforming an arbitrary classical block compression algorithm into a quantum compression algorithm and provide a new, simplified argument for the existence of universal quantum compression algorithms.

1. Motivation

Why is quantum compression important? The essential reason that quantum compression might be important is the cost of qubits. In particular, various experiments with quantities limited to sizes such as "3 qubits" and "5 qubits" suggest that it is intrinsically difficult to work with (and store) a large number of qubits.

Working on the compressed form of a quantum state is probably at least as difficult and impractical as running a classical algorithm on compressed state although the tradeoffs may differ. It is unclear how to compare the cost of quantum operations to the cost of storing quantum state. As the ratio (quantum operation cost)/(qubit storage cost) becomes smaller, the process of quantum storage becomes more intrinsically attractive.

Quantum compression is a technique which will only work approximately. Guaranteeing that the approximation is 'tight enough' will require that quantum compression only be executed on a large number of qubits - at least 10^2 or 10^3 in order to gain significant wins. This is one or two orders of magnitude beyond the amount of state used by current quantum computers which suggests that it will be awhile before quantum compression is used in any computational devices.

Date: 5/7/01.

Another possible use is in quantum communication. In particular, if Alice wants to send 10^3 qubits to Bob, Alice might first compress the 10^3 qubits into $5 * 10^2$ qubits and transmit them to Bob. Bob could then decompress the $5 * 10^2$ qubits into 10^3 qubits and use them as if they were Alice's original 10^3 qubits.

2. Quantum Compression Setting

2.1. **Setting.** Our basic setting will consist of a quantum source with some distribution, denoted by p_i, over quantum states $|a_i\rangle$ which we wish to compress into new quantum states $|b_i\rangle$ with the property that the $|b_i\rangle$ exist in a lower dimensional Hilbert space then the original $|a_i\rangle$. Restating, our quantum source emits vectors with the following density matrix:

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

Here the $|\phi_i\rangle$ are the eigenvectors of the density matrix and q_i are the eigenvalues.

2.2. **Measuring success.** The notion of "success" is somewhat different between classical compression and quantum compression. For classical compression, the compression was succesful if the initial state was recovered. The quantum mechanical analog will be "Success if we compare the input and output quantum states and they are the same". If $|a_i\rangle$ is our input and our output is $|a_i'\rangle$, then the fidelity will be defined by:

$$F = \sum p_i |\langle a_i | a_i' \rangle|^2$$

If all of the involved states are orthogonal, then this notion will be equivalent to the "success probability" from classical compression. There is an important difference in the nonorthogonal case: we allow small errors in the Hilbert space vectors. This implies, for example, that in addition to occasionally getting things "very wrong", we allow ourselves to often get a state which is "slightly wrong" in the sense that it isn't quite the same as the original. This notion of fidelity is fairly natural although it has some implications: if we are relying on the exact value of a very entangled quantum state (such as in the early stages of Grover's algorithm), it is unclear that this notion of fidelity will be good enough. Grover's algorithm works by a process of iterated transformations which magnify small amplitudes. Since the fidelity is not greatly altered by the change of a low amplitude eigenvector, the state of Grover's algorithm might be significantly damaged if it went through a compression/decompression cycle.

2.3. **Important quantum effects.**

2.3.1. *No-cloning.* The first and most important difference between a quantum compression and classical compression is the no-cloning theorem. The no-cloning theorem [11] says that it is impossible to copy a quantum state from any hilbert space of dimension larger then 1. In particular, the following transformation is forbidden by the linearity of quantum mechanics:

$$|a0\rangle \rightarrow |aa\rangle$$

On the face of it, this is a very depressing result since it appears that we can not communicate. However, there is an alternative 'forgetting' notion of communication which is allowed by the laws of quantum mechanics. This transformation is:

$$|a0\rangle \rightarrow |0a\rangle$$

Thus, it is possible to send quantum information as long as it is forgotten at the source. This is a very important restriction which implies any quantum communication scheme must be very careful about which information it sends.

2.3.2. *Entanglements: No measurements.* There are limitations imposed by the nature of quantum computation which make compression inherently more difficult. In particular, every measurement of the quantum state must be considered very carefully. This implies that standard quantum algorithms must run "blindly" with operations undertaken on unobserved state. Interestingly enough, the notion of "blind execution" arises in the planning literature under the name "conformant planning" [9]. Results from conformant planning can not directly apply to the analysis of a quantum systems because conformant planning does not take into account the linearity of quantum operations. Nonetheless, it is interesting to see the concept arise elsewhere.

Since we can not measure the result of a quantum computation, asynchronous approaches to computation [10] have inherent difficulties. The asynchronous approach is not entirely precluded by quantum mechanics but it appears difficult to implement efficiently since it is necessary for operations to take the same amount of time independent of their state and to signal their completion without providing evidence of their state. No technique is known for signalling completion in a quantum mechanical method. The implication of this is that we do not know how to create a quantum compression/decompression algorithm which works on demand. It is also not possible to use a compression algorithms which run in time dependent upon the quantum state. When undertaking quantum compression, we must predict the running time of each step in the quantum compression without measuring the quantum state we desire to compress.

This is also true of "online algorithms" [1] which process and emit information dynamically while changing their internal state adaptively. For an online algorithm, the output state would become entangled with the internal state of the algorithm resulting in spurious entanglements. Consequently, quantum compression/decompression algorithms which work in an online fashion to improve their compression ratio will not work.

We are left with only the notion of a synchronous algorithm with carefully planned measurements. In particular, This is in strong contrast to many (perhaps even the majority) of classical compression algorithms in use today which often fall into the "online algorithms" setting in practice. For example, the commonly used "gzip" program has a running time which is dependent upon the nature of the data being compressed.

2.3.3. *Unknown compression ratio.* Can we cope with an unknown compression ratio? Most classical compression algorithms can cope with unknown and even varying compression ratios. In the quantum compression case, this is difficult because the running of the (compression, decompression) algorithm is dependent on the achieved compression ratio which may not be known *a priori*.

The answer is “sort of”. We could imagine augmenting the output of a quantum compression algorithm with some extra qubits which are used to store some information about the compression ratio. The compression ratio will be measured before the decompression and used in deciding which exact decompression algorithm should be used (and how many qubits it should be applied to). In order to work, the value of these qubits *must* be the same for most (as measured by the fidelity) of the quantum states in the initial ensemble. Thus, it is possible to cope with varying compression ratios as long as the following property exists:

Almost all elements in the initial ensemble has the same compression ratio

This is a very severe restriction which is almost (though not quite) the same as disallowing variable compression ratios.

3. Basic Quantum Compression

Quantum information theory is essential to the notion of quantum compression. Quantum information theory provides a lower bound on the number of qubits which must be used in order to encode a quantum state.

3.1. The noiseless coding theorem.

This theorem by Schumacher [8] is strongly analagous to the information theory results of Shannon.

Theorem 3.1. *(quantum noiseless coding theorem) Let M be a quantum signal source with a density matrix $\rho = \sum_i p_i |a_i\rangle\langle a_i|$. Then, for any $\epsilon, \delta > 0$, the quantum signal can be compressed and decompressed using an average of $S(\rho)+\delta$ qubits where $S(\rho) = \rho \log \rho$. The fidelity of the decompressed state with respect to the original state is at least $1 - \epsilon$.*

Proof. Detailed in [8] although part of it is discussed in the next subsection. □

The noiseless coding theorem for quantum information theory is a generalization of the noiseless coding theorem for classical information theory because $S(\rho) = H(p_i)$ when the individual $|a_i\rangle$ are orthogonal. However, the value of $S(\rho)$ decreases as the individual $|a_i\rangle$ become less orthogonal and $S(\rho) = 0$ when all of the $|a_i\rangle$ are the same. This is physically plausible because the state of the system can not be distinguished by any measurement.

The noiseless coding theorem also works for systems composed of two entangled subsystems, $M + R$. In particular, our density matrix ρ might arise from the partial trace, $\mathrm{Tr}_R(|a_{M+R}\rangle)$ of some highly entangled pure system. Then, when we transfer the state of ρ to another system, ρ', we would hope that the entanglements between the rest of the system and the qubits of ρ' are (approximately) the same as the entanglements between the rest of the system and the original ρ. This is the case as can be seen by a careful analysis of the schmidt decomposition as in [8]. The carryover of entanglements is a very important point since it allows for the possibility of the development of a distributed quantum computer.

3.2. **Realizing the quantum noiseless coding theorem.** One direction (realizability) of the quantum noiseless coding theorem is proved by construction, giving us our first code. In particular, Schumacher exhibited a compression/decompression scheme which used $N(S(\rho)+\delta)$ qubits to encode N examples of a system with $S(\rho)$ entropy achieving a fidelity of $1-\epsilon$. Here ϵ and δ are two numbers which can be made arbitrarily small as N increases.

The basic scheme uses block coding and suffers from the drawbacks of the Shannon noiseless coding theorem: it is impractical for real use.

The basic approach is to note that our qubits will almost entirly lie within a Hilbert space of size $2^{N(S(\rho)+\delta)}$. Consequently, we can rotate the largest $2^{N(S(\rho)+\delta)}$ eigenvectors into this Hilbert space without loss and truncate the components not in the smaller Hilbert space. Then, decompression is done by using the reverse rotation in a space concatenated with a sufficient number of $|0\rangle$ qubits.

The proof that this works is detailed in [8], but it essentially works by proving that almost all vectors will have components mostly in the large eigenvalue directions.

Further work [6], has produced an explicit algorithm with n^3 running time in the case of a qubit source. Since we are dealing with a qubit source, there are only two eigenvectors of the density matrix, ρ. One of these eigenvectors, the principle eigenvector, will have a larger eigenvalue then the other eigenvector. Let's denote the eigenvectors by $|0\rangle$ and $|1\rangle$ where $|0\rangle$ has a larger eigenvalue then $|1\rangle$. Then, in the eigenvector basis, the vectors we wish to compress will have the form, $\sum_i c_i |0010011000010011001000100\rangle$. The algorithm works by calculating the index of an eigenvector in a space of size $2^{n(S(\rho)+\delta)}$ based upon the number and location of principle eigenvectors in the block.

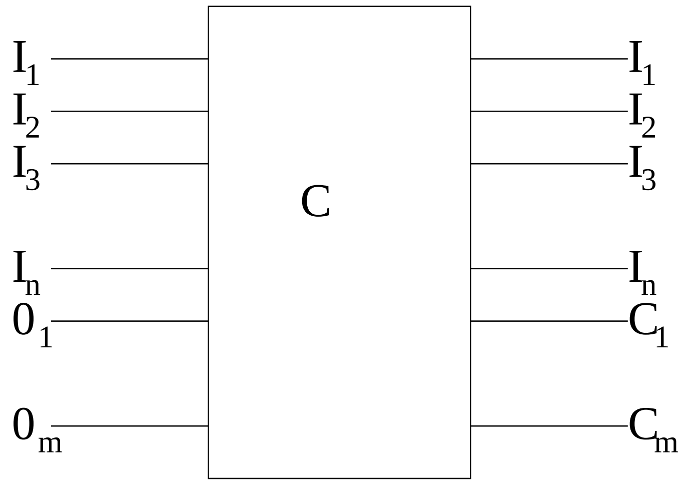
This brings up one important point: most quantum compression papers work in the eigenvector basis. This does not change the asymptotic computational complexity since transforming between the eigenvector basis and the computational basis takes n rotations (1 per qubit) and all of the compression algorithms will require more then $O(n)$ operations.

Another important detail is that we *require* knowledge of which rotation/unrotaton we can accomplish. We could attempt to make a more adaptable coding algorithm by encoding the rotations in the compressed form. This will be further discussed later.

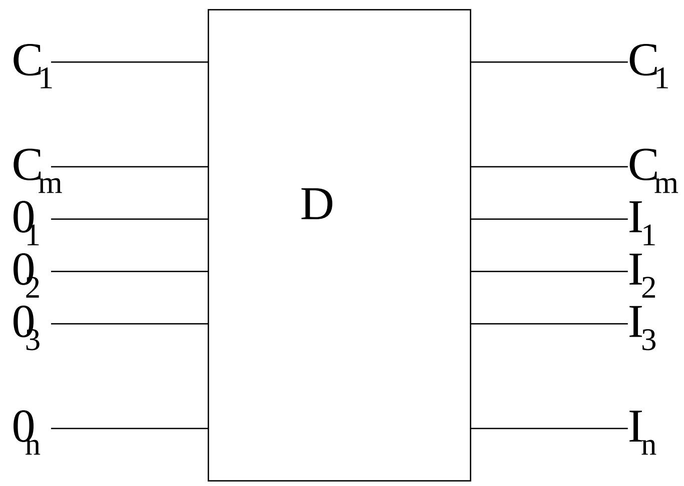
4. Quantum compression extensions

4.1. **Generic Compression.** A generic notion of compression can be derived based upon the idea that any function can be executed on a quantum computer.

In particular, a generic (classical) compression algorithm, $C : \{0,1\}^n \to \{0,1\}^m$, maps n bits to m bits with n larger then m. Our decompression algorithm, $D : \{0,1\}^m \to \{0,1\}^n$, will then (hopefully) invert the compression to give the original state. Now, we can use the general result that any classical function can be implemented [2] (reasonably efficiently) on a quantum computer. This allows us to compute the following (reversible) compression circuit:

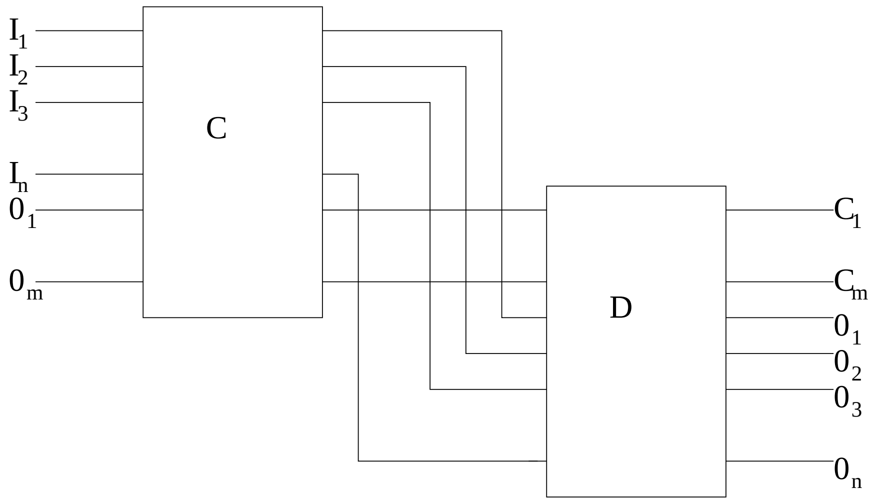


Similarly, we can construct the following reversible decompression circuit which takes compressed input and creates uncompressed output:



These circuits have a special relationship with respect to each other because the decompression circuit *is* the inverse of the compression circuit *if* the compression/decompression cycle is lossless. This "if" implies that while a map from n bits to m bits (for $m < n$) is not in general unitary, we can hope that compression will somehow be doable in a "natural" manner.

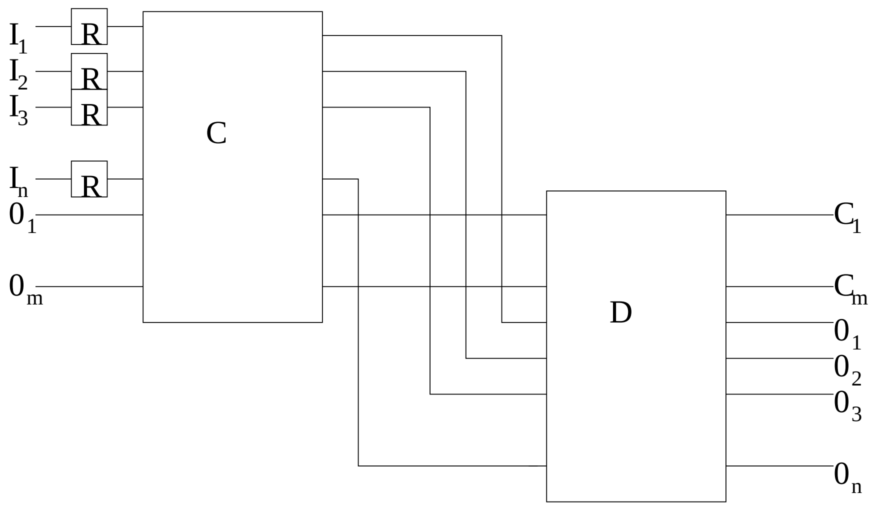
To see this, note that we are not finished with the compression because our output leaves the compressed bits entangled with the input bits (with all attendant undesirable side effects). There is a simple construction which uses the decompressor to remove the entanglement left in the output of the compressor.



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 $I_i \oplus D(C(I))_i$ rather then $0 \oplus D(C(I))_i = D(C(I))_i$. This term is 0 precisely when the compression is lossless. At this point, we can measure the all of the "0" qubits. If any of our measurements are nonzero, we have detected a low probability event. Otherwise, we can send the compressed qubits over some quantum channel.

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

This approach only does *classical* compression of qubits. In particular, it will fail to compress some highly compressible state distribution such as the state which is $(|0\rangle + |1\rangle)^k$ 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^k (Hadamard applied to each bit), would convert the input to $|0^k\rangle$ which many compression schemes would greatly compress. As pointed out previously, this extra rotation involves $O(n)$ operation since it can be done on each symbol from the quantum source individually. This gives us the following circuit:



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 [4]. 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 $S' = S \log T$ and the reversible time is (for any ϵ) $T' = \frac{1}{\epsilon} S T^{1+\epsilon}$. This implies that a block compression algorithm such as bzip which runs in $S = n$ space and $T = n \log n$ time will require $S' \simeq n \log n$ space and $T' = \frac{1}{\epsilon} n (n \log n)^{1+\epsilon}$ time. This is surprisingly competitive with other results considering it is compiled rather then hand built. It is easy to imagine that hand compilation of irreversible compression/decompression algorithms would yield further functional improvements.

4.2. **Huffman compression.** Several people [3] have placed some thought into removing some of the limitations of quantum codes as discussed in section 2.3.2. They do not succeed in making an 'instantaneous' quantum code although they do employ a quantum analog of the huffman code to achieve several speedups and a bit more flexibility then the naive block code approach:

(1) Parallel quantum gate count of $n(\log n)^3$
(2) Sequential quantum gate count of $n^2(\log n)^3$
(3) A procedure for halting the communication of compressed qubits with little loss in fidelity and a small amount of extra communication.

4.3. **Arithmetic compression.** Some other people [5] have worked out a very explicit arithmetic compression scheme for a quantum source which requires n^3 running time.

The authors expend a significant amount of effort on first projecting onto the "typical subspace" before running the arithmetic compression algorithm. I believe this is unnecessary - instead, we can simply assume that our vector falls in the "typical subspace" and measure the $|0\rangle$ outputs of figure 9 in order to achieve a projection into the typical subspaces. If the result of the measurements is not a $|0\rangle$ vector, then we are not in the typical subspace. Otherwise, we are necessarily in a typical subspace because only elements in the typical subspace will have $|0\rangle$ vectors.

4.4. **Universal quantum compression.** In classical compression, there is a notion of "universal compression" which is essentially asymptotically optimal compression *without* knowledge of the entropy of the source. The noiseless coding theorem relies upon information which may be difficult to acquire: the density matrix of the source. Somewhat surprisingly, this is information is similarly eliminatable. We can design a code which achieves asymptotically optimal quantum compression without knowledge of the density matrix, ρ.

There is a simpler, tighter argument for this then the one used in [7]. Assume that our quantum source is d-dimensional. For example, a qubit source would be $d = 2$ dimensional. Then, there are d eigenvectors to this d-dimensional source and each eigenvector will be specifiable with a d-dimensional vector. Thus, we can overestimate the number of free parameters at d^2 as in [7].

However, there is no need to overestimate. In particular, note that the first eigenvector has $d-1$ free dimensions since all parameters are free up to a scaling constant. The second eigenvector will have $d-2$ free dimensions since all parameters are free up to a scaling constant and orthogonality with the first eigenvector. Continuing this process we get that the number of free parameters is:

$$\sum_{i=0}^{d-1} i = \frac{d(d-1)}{2}$$

We will wish to use a code which first specifies the eigenvectors and then uses the standard encoding procedure given the eigenvectors. There are two critically important facts which imply that the process of first specifying the eigenvectors results in negligible overhead:

(1) The precision with which we must specify the eigenvectors in order to maintain constant fidelity is $O(n)$ (the number of source symbols in the code).
(2) The cost of specifying the eigenvectors to $O(n)$ precision is $O(\frac{d(d-1)}{2}\log n)$ *bits* (qubits are unnecessary).

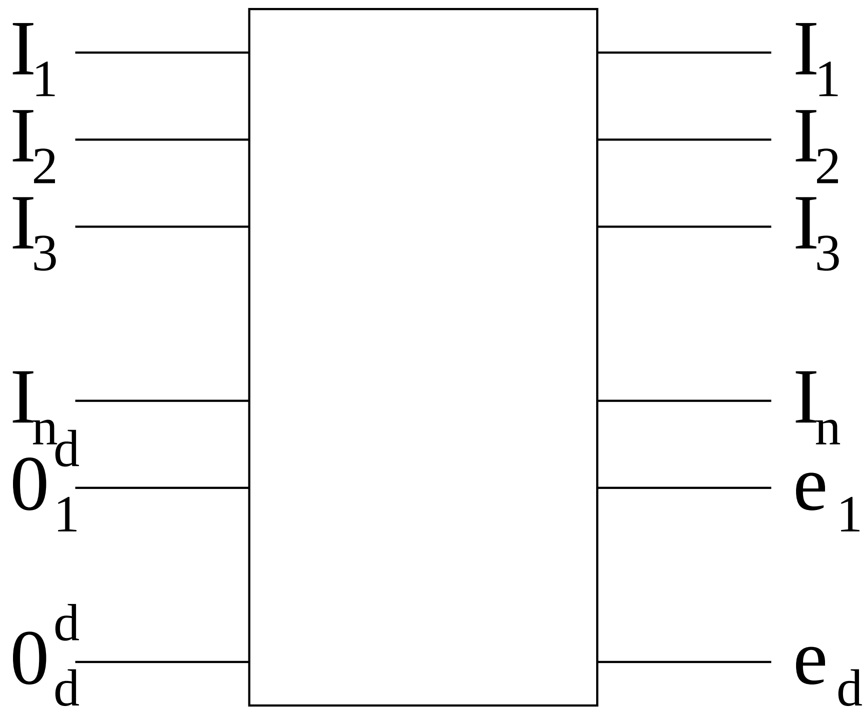
Since only $O(\log n)$ bits are required,

$$\lim_{n\to\infty} \frac{\text{cost of eigenvectors}}{\text{minimum cost of encoding}} = \lim_{n\to\infty} c\frac{\frac{d(d-1)}{2}\log n}{(S(\rho)-\delta)n} = 0$$

which implies that the cost of specifying the eigenvector bits is a negligible part of the total.

Therefore, a universal quantum encoding can be specified as a (eigenvectors (in bits), encoding (in qubits)) pair with the eigenvectors asymptotically smaller then the encoding. Decoding is as in nonuniversal encodings with the final rotation specified by the eigenvectors.

Although it is clear what a universal quantum encoding is, it is unclear what the most efficient technique for realizing this encoding is. The simplest approach is to measure the source $O(n^2 \log n)$ times and estimate the density matrix. Given the density matrix, we can calculate the eigenvectors based upon these measurements. While this is the simplest approach, it is not ideal because it involves many measurements. A better technique would be a quantum algorithm which deduces the eigenvectors of the source so we could transmit the eigenvectors to the destination without losing source symbols to measurement. In essence, we want to use the following circuit:



This circuit takes as input the source symbols to be encoded and outputs the eigenvectors of the source. It seems difficult, to realize this circuit and it may simply be impossible to realize. Thus, while it is feasible to have a universal *encoding,* the feasibility and construction of a universal *encoder* is an open problem.

Universal quantum compression is not quite as "universal" as universal classical compression. In particular, in the classical case we do *not* need to know $H(P)$, a bound on the Shannon entropy of our source while in the quantum mechanical case, we must know $S(\rho)$ in order to achieve asymptotically optimal compression. This weakness is related to the difficulties introduced in section 2.3.2 which disallow quantum compression with an unknown length.

5. Conclusion

Quantum compression and decompression is moderately well understood and significant work remains to be done. Our understanding consists of a lower bound on quantum coding efficiency and matching codes which asymptotically achieve the lower bound. However, there is much left to investigate, including:

(1) To what extent can we overcome the limitations discussed in section 2.3.2 and 2.3.3?
(2) How efficiently can we do universal quantum source coding?
(3) How can we realize universal quantum source coding with a quantum circuit?
(4) How efficiently can we even do quantum source coding?
(5) What about nonmemoryless quantum sources?

And, of course, there is the simple question of application. So far, no quantum machine yet exists with enough state to efficiently take advantage of quantum compression. If this changes in the future, it will strongly motivate further investigation in quantum compression.

References

[1] Avrim Blum, course, http://www.cs.cmu.edu/~avrim/Approx00/
[2] Ethan Bernstein and Umesh Vazirani, "Quantum complexity theory", 25th Association for Computing Machinery Symposium on the Theory of Computing, http://citeseer.nj.nec.com/bernstein93quantum.html

[3] Samuel L. Braunstein, Christopher A. Fuchs, Daniel Gottesman, and Hoi-Kwong Lo, "A quantum analog of Huffman coding", 1998 IEEE Symposium on Information Theory, p. 353 http://xxx.lanl.gov/abs/quant-ph/9805080

[4] Harry Buhrman, John Tromp, Paul Vitanyi, "Time and Space Bounds for Reversible Simulation", Proc. International Conference on Automata, Languages and Programming, Lecture Notes in Computer Science, Vol ??, Springer-Verlag, Berlin, 2001. http://www.cwi.nl/~paulv/papers/tradeoff.ps

[5] Isaac L. Chuang and Dharmendra S. Modha, "Reversible Arithmetic Coding for Quantum Data Compression," IEEE Transactions on Information Theory, vol. 46, no. 3, pp. 1104-1116, May 2000. http://www.almaden.ibm.com/cs/people/dmodha/qcode.ps

[6] Richard Cleve and David P. DiVincenzo, "Schumacher's quantum data compression as a quantum computation", Phys. Rev. A54, 4707, (1996), http://xxx.lanl.gov/ps/quant-ph/9603009

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

[8] B. Schumacher, "Quantum Coding" Physical Review A, Vol. 51, pp. 2738-2747, 1995 http://citeseer.nj.nec.com/rd/89477878%2C105399%2C1%2C0.25%2CDownload/http%253A//citeseer.nj.nec.com/compress/0/papers/cs/411/http%253AzSzzSzvesta.physics.ucla.eduzSz%257EsmolinzSzpaperszSzqubit.ps.gz/schumacher95quantum.ps

[9] D. Smith and D. Weld, "Conformant Graphplan" Proceedings of the Sixteenth National Conference on Artificial Intelligence, Madison, WI, 1998 http://ic-www.arc.nasa.gov/ic/people/de2smith/publications/cgp.ps

[10] Asynchronous VLSI group at Caltech. http://www.async.caltech.edu/

[11] W. K. Wooters and W. H. Zurek, Nature 299, 802 (1982).