David Gibson
Vazirani Class #??
October 30, 1997

Quantum Turing Machines.

1 Simple Quantum Turing Machines

We would like to have a formal model for a quantum Turing machine, which is as close as possible to a classical Turing machine, but which obeys the constraints of unitarity. In particular, a QTM should be reversible.

The machine is defined over a state set Q, and an alphabet Σ of symbols, which appear on a single infinite tape, accessed by a single head. It is specified by a unitary transformation $M : \mathcal{C}^{Q\times\Sigma} \rightarrow \mathcal{C}^{Q\times\Sigma}$ and a map $f : Q \rightarrow \{L, R\}$. Let the state of the machine be q and the state of the tape at the current position be σ. Each time step consists of two operations:

Change state and tape M maps q, σ to q', σ'.

Move head The tape head moves according to $f(q')$.

Note that both operations are reversible, and since the direction of motion depends only upon the *new* state, the machine is reversible.

Example. Let $Q = \{p, q\}$ and $\Sigma = \{0, 1\}$. If M is such that $M|p, 0\rangle = \frac{1}{\sqrt{2}}(|q, 0, L\rangle + |q, 1, L\rangle)$, and $M|p, 1\rangle = \frac{1}{\sqrt{2}}(|q, 0, L\rangle - |q, 1, L\rangle)$, (and $f(q) = L$, which has been absorbed into the above notation), the machine will perform a Hadamard transform on the first tape cell, move left, and perform whatever action is specified for state q.

The evolution operator of the entire machine can be visualized as a countably-infinite dimensional unitary matrix, U, operating on the vector of the amplitudes of all possible configurations, v. A configuration is specified by q, σ, a head position h, and state information S for all the tape cells, excluding the current cell. If we order v so that all the $|Q \times \Sigma|$ configurations corresponding to a particular h and S are contiguous, then we can write $U = F \begin{pmatrix} M & & \\ & \ddots & \\ & & M \end{pmatrix}$.

The rightmost matrix performs the state change and tape write, and F is a permutation matrix which maps a particular (h, S, q, σ) amplitude to some other (h', S', q, σ'), depending on the value of $f(q)$. Clearly then, U is unitary.

2 A Universal Quantum Turing Machine

We can use our QTM model to build a universal QTM, $\mathcal{U}$, in much the same way as in the classical case. Once again, unitarity introduces a few difficulties.

Since the simulated machine may have a considerably larger alphabet than $\mathcal{U}$, the input tape will contain some *encoding* of the input for the QTM to be simulated, where each simulated cell may correspond to a block of several $\mathcal{U}$-cells, along with some encoding of the M and f maps for the simulated machine.

Since simulated cells are mapped onto blocks, any transformation which puts a cell into a superposition of states has to do so *atomically*: the state $(|a\rangle + |b\rangle) \otimes (|c\rangle + |d\rangle)$ (which would be contructed if we proceeded naïvely, a symbol at a time) is not the same as $|ac\rangle + |bd\rangle$. Further, when constructing superpositions, we are constrained by reversibility: there must be a one-to-one mapping from initial to final states, which means we cannot leave any extra bits lying around after the transformation. We must build exactly what is required, and no more.

$\mathcal{U}$will alternate between applying the M transformation, and using f to move the head. We shall detail the M transformation.

Encoding M. The first issue is implementing an arbitrary unitary transform. We can factor any unitary matrix M as $\Pi_j^{poly(|Q\times\Sigma|)} U_j$, where the U_j are of the form:

$$\mathbf{A:}\ U_j = \begin{pmatrix} \ddots & & & & \\ & 1 & & & \\ & & e^{i\theta} & & \\ & & & 1 & \\ & & & & \ddots \end{pmatrix} \text{ or } \mathbf{B:}\ U_j = \begin{pmatrix} \ddots & & & & \\ & 1 & & & \\ & & R_\phi & & \\ & & & 1 & \\ & & & & \ddots \end{pmatrix},$$

where R_ϕ is a 2×2 rotation matrix.

Then, to apply M, we apply each element of this factorization in turn. Here we need to build atomic cell operations. What we require is a deterministic one-to-one "unpacking" function which maps a value $|v\rangle$ into a tuple $|v\rangle|v'\rangle|b\rangle$, where v' is the state we wish to transform v into, and b is a (single $\mathcal{U}$-cell) flag which indicates which of the two preceding states is "intended".

The operation then consists of unpacking, performing a unitary transform on b, and repacking. For type A, $v' = v$, unless v indicates the special row, in which case v' is v with rotated amplitude, and the operation on b simply flips the intended location.

For type B, $v' = v$ unless v is one of the two R_ϕ rows, in which case v is set to the first row, v' to the second, and b accordingly. A rotation on b performs the atomic unitary operation, and repacking generates the appropriate rotated state.

With this encoding, we can perform the rest of the operations deterministically and reversibly. We will ensure that at the start of each cycle, there is an empty workspace on the tape. First, we examine the current q and σ, and use the encoding of M, described below, to generate q' and σ' in this workspace. In order for this to be reversible we do this by some form of exclusive-or'ing process. Then we exchange the values of q, σ and q', σ' (which is again a reversible operation). Lastly, we use an encoding of M^{-1} to erase the q, σ from the workspace (by XOR-ing over it). The new q', σ' are in place, and we are ready to begin again.

3 Removing the 'simple' restriction

The 'simple' machine description given above manages to exclude some machines which still undergo unitary evolution.

Example. The following still describes a unitary evolution, even though f is not a function.

$$M|q,0\rangle = \frac{1}{2}(|p,0,L\rangle + |q,0,L\rangle + |p,0,R\rangle - |q,0,R\rangle$$

$$M|p,0\rangle = \frac{1}{2}(|p,0,L\rangle + |q,0,L\rangle - |p,0,R\rangle + |q,0,R\rangle$$

However, if we let $p' = p + q$, $q' = p - q$, then f becomes single-valued again. In general, we have the theorem:

> For a single-tape QTM, there exists some basis for $\mathcal{C}^Q$ under which f is single-valued.

Thus, our universal machine need only convert the state distribution into this basis, run the simulation, and convert back.