CS 880: Quantum Information Processing 9/9/2010

Lecture 3: Constructing a Quantum Model

Instructor: Dieter van Melkebeek Scribe: Brian Nixon

This lecture focuses on quantum computation by contrasting it with the deterministic and probabilistic models. We define the model and discuss how it benefits from quantum interference and how observations affect the quantum behavior. Then we talk about the smallest sets of gates that can be used to generate the quantum model and approximations of that model (after defining an approximation). This should set us up to discuss quantum algorithms in the near future.

1 Defining the Quantum Model

We're building a model of quantum computing as a set of linear operations on a register of qubits. We have previously done this for deterministic and probabilistic models of computation; here we compare them to get our desired model.

1.1 Probabilistic Model

As shown last time, the probabilistic model consists of

1. State: We conside the data register to exist in a "pure" state - a superposition $|\psi\rangle = \sum_s p_s|s\rangle$ where $\sum_s p_s = 1$ and $\forall s$, $0 \leq p_s \leq 1$. Each $|s\rangle$ is the representative for a given bit vector.
2. Operations:
   (a) Local probabilistic gates T. These can take the form of stochastic matrices.
   (b) Observation. Look at the register and see what bit vector it contains. Can think of as a map $|\psi\rangle \rightarrow |s\rangle$ with likelihood p_s.
3. Uniformity issues. The gates have to be described concisely. Refer to the last lecture but can be imposed by using a deterministic Turing machine to control sequence of gates.

We can ask what effect observation has on the computation process. While the result after an observation is always a single bit vector, it is useful to treat it abstractly by considering the likelihood of all possiblities. This leads to the following definition.

Definition 1. *A "mixed" state - a probability distribution over a set of pure states Given a pure state,* $\text{obs}(\sum_s p_s|s\rangle) = \{(|s\rangle, p_s)\}_s$ *where our notation describes the set of possibilities as tuples of the state and its likelihood.*

We can run an algorithm on a mixed state by using $|s\rangle$ as the input and multiplying the output by the likelihood, p_s.

Exercise 1. *Prove that intermediate observations in the probabilistic model can be suspended in favor of one observation at the end without altering the likelihood of each output.*

1.2 Quantum Model

Using this as a template we can describe the quantum model of computing similarly.

1. State: The "pure" state - single superposition $|\psi\rangle = \sum_{s\in\{0,1\}^m} \alpha_s|s\rangle$ where $\forall s, \alpha_s \in \mathbb{C}$ and $\sum_s |\alpha_s|^2 = 1$. We can also consider the register to be in a mixed state.

2. Operations:

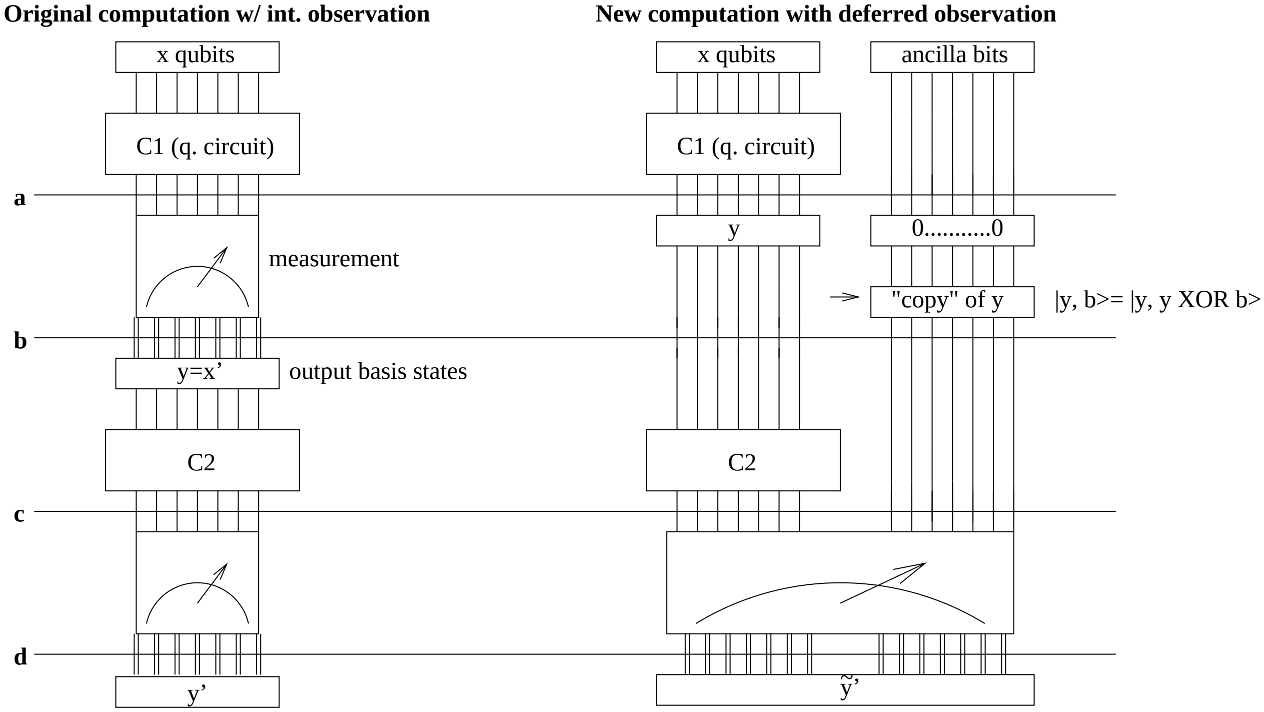
   (a) Local unitary operations. These take the form of unitary matrices (i.e. $T^*T = I = TT^*$).

   (b) Observation. Collapses $\sum_s \alpha_s|s\rangle$ to a distribution where $|s\rangle$ occurs with probability $|\alpha_s|^2$. Unlike the probabilistic model, in the quantum model intermediate observations will affect the final distribution. Consider the following:

3. Uniformity conditions that we'll talk about later.

Example: Let H be the Hadamard matrix $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. We saw last lecture that $H^2|x\rangle = |x\rangle$. Now $\text{obs}(H|x\rangle) = \{(|0\rangle, \frac{1}{2}), (|1\rangle, \frac{1}{2})\}$ where our notation for the mixed state matches that of the definition above. $H(\text{obs}(H|x\rangle)) = \{(\frac{|0\rangle+|1\rangle}{\sqrt{2}}, \frac{1}{2}), (\frac{|0\rangle-|1\rangle}{\sqrt{2}}, \frac{1}{2})\}$ not $|x\rangle$. The intermediate observation has a distinct affect in our output. ⊠

Just as interference is lost by making observations in the quantum model, interference can also be gained by failing to make an observation step. In particular, this affects composition of quantum machines. We can remove the necessity of these intermediate observation steps if we use CNOT to copy the output after each subfunction terminates. Consider the following diagram. Here C1 and C2 are the machines we want to compose. By saving the state of the register after C1, we remove the potential interference. Subsequent observation of that register state is not necessary so long as it exists in a state where it can be observed.

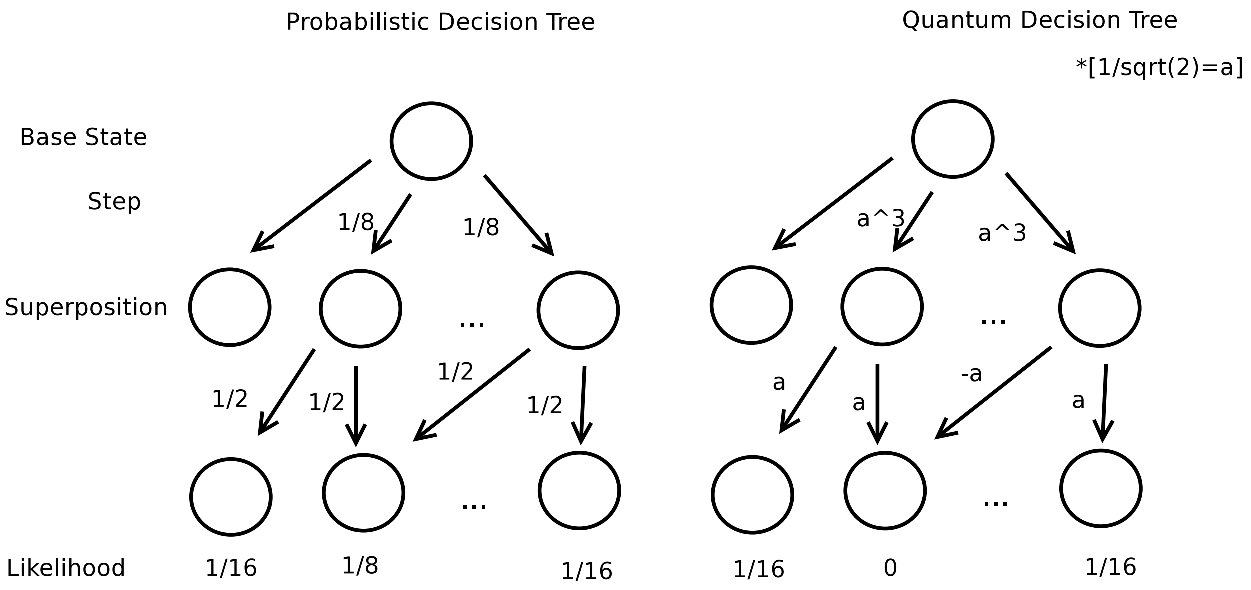


As discussed in the last lecture, simulating the deterministic model using the quantum model by making the algorithm reversible requires only a little more resources. If we make an observation after every step in the quantum model we can simulate any probabilistic machine. Of course, this sacrifices the destructive interference we obtained by using the phase information in the quantum model.

1.3 Uniformity

Certain conditions have to be placed on our circuits to make them physically realizable, we call these the uniformity conditions. We defined our uniformity conditions in the deterministic model by using Turing machines to describe behavior (or the equivalent circuit model). For the probabilistic model we imposed the restriction that any "coin flip" choices had to be done with a fair coin. For quantum machines we can impose two conditions. First, we specify that the gates used need to be formed in polynomial time using a deterministic Turing machine. We could use a quantum Turing machine instead but would require some orthogonality conditions to restrict the transition function that we're not going to go into. Second, the amplitudes in our circuits can't be too complicated. While theoretically we have access to all unitary matrices, practically there should be a limit to the degree of precision we can use in generating the matrix entries. It is not probable that a randomly selected complex number can be generated from a base set using simple operations, however it is possible to get close while maintaining restrictions on the number and type of operations used. If we allow arbitrary precision we can encode objects, such as the characteristic sequence of the halting function, which will cause large problems.

We should note that the power of quantum computing doesn't come from there somehow being more computational paths that it can take but from the interference between these paths. If you look at the possible state diagrams of a quantum machine vs. a probabilistic machine, they are formed on the same decision tree over the register. However, the coefficient types defining how probable a given state is are different.



2 Universal Sets

Uniformity can be more formally expressed using universal sets of gates.

Definition 2. *S is an* exact universal set *for a model if any gate from that model can be realized exactly using only combinations of gates from S.*

For the deterministic model, {NAND} is an exact universal set (i.e. any deterministic gate can be composed of NAND gates operating on the correct bits). For the probabilistic model, $\{\text{NAND}, \text{COIN-FLIP}_p\}$ generates an exact universal set where COIN-FLIP_p is a coin flip operation on a single bit operating with bias p.

Exact universal sets require a degree of precision that is not always helpful when trying to physically realize circuits for a model. We can relax the constraints to get sets that are "good enough" for general purposes.

Definition 3. *S is an* universal set *for a model if any gate from that model can be realized arbitrarily precisely using only combinations of gates from S.*

For practical reasons, we don't want our generating set to be infinite and so try to get finite universal sets to approximate our model. For example, for the probabilistic model, $\{\text{NAND}, \text{COIN-FLIP}_{1/2}\}$ is a universal set and, being finite, is better suited for circuit realization that the exact universal set. For the quantum model, $\{\text{CNOT}, \text{SINGLE-QUBIT-GATE}\}$ generates the exact universal set where SINGLE-QUBIT-GATE refers to any unitary operation on a single qubit (i.e. any 2×2 unitary matrix). Every local unitary operation on a constant number of qubits can be formed by a composition of these operators. However, this too is an infinite set and we would like to be able to approximate its behavior using a finite generating set.

It is necessary to be formal when we talk of approximating. Given a fixed error term $\epsilon > 0$ we want a generating set that lets us get close to any operation of a fixed complexity with error less than ϵ. If U denotes the exact unitary transformation and $\tilde{U}$ an approximation, we want it to be the case that for each initial state $|\psi\rangle$, the probability distributions obtained by observing $U|\psi\rangle$ and $\tilde{U}|\psi\rangle$ are at most ϵ apart, i.e., for $p = \mathrm{obs}(T|\psi\rangle)$ and $\tilde{p} = \mathrm{obs}(\tilde{T}|\psi\rangle)$, $\mathrm{D}(p,\tilde{p}) = ||p-\tilde{p}||_1 = \sum_s |p_s - \tilde{p}_s| \leq \epsilon$.

The sequence of unitary operations before the final observation gives rise to an overall unitary transformation that is the product of unitary transformations U of the form $U = T_i \otimes I_m$, where T denotes an elementary quantum gates. Letting $\mathrm{D}(U,\tilde{U}) = ||U - \tilde{U}||_2$ it is left as an exercise to show the following:

Exercise 2. *Prove the relations.*

- $\mathrm{D}(p,\tilde{p}) \leq 2\,\mathrm{D}(U,\tilde{U})$.
- $\mathrm{D}(U_t U_{t-1} ... U_1, \tilde{U}_t \tilde{U}_{t-1} ... \tilde{U}_1) \leq \sum_{i=1}^{t} \mathrm{D}(U_i, \tilde{U}_i)$.
- $\mathrm{D}(T \otimes I_m, \tilde{T} \otimes I_m) = \mathrm{D}(T,\tilde{T})$.

Note that $||A||_2 = \max_{x \neq 0}(\frac{||Ax||_2}{||x||_2})$ which is equivalent to the square root of the maximal eigenvalue of A^*A. These relations tell us if our operations are restricted to be combinations of at most t elements of S then to get error of ϵ it suffices to show $||T_i - \tilde{T}_i||_2 \leq \epsilon'$ where ϵ' is taken to be $\frac{\epsilon}{2t}$.

The following theorems have their proofs omitted.

Theorem 1. *For any universal set S, $\forall \epsilon > 0$ we can approximate any T to within ϵ by a combination of* poly-log$(1/\epsilon')$ = poly-log$(2t/\epsilon)$ *gates from S and S^-1 (any gates from S and their inverses).*

Theorem 2. $S = \{\mathrm{CNOT}, H, R_{\pi/4}\}$ *is universal. Here H is the Hadamard matrix used above and* $R_\theta = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$.