CS290A, Spring 2005:

Quantum Information & Quantum Computation

Wim van Dam

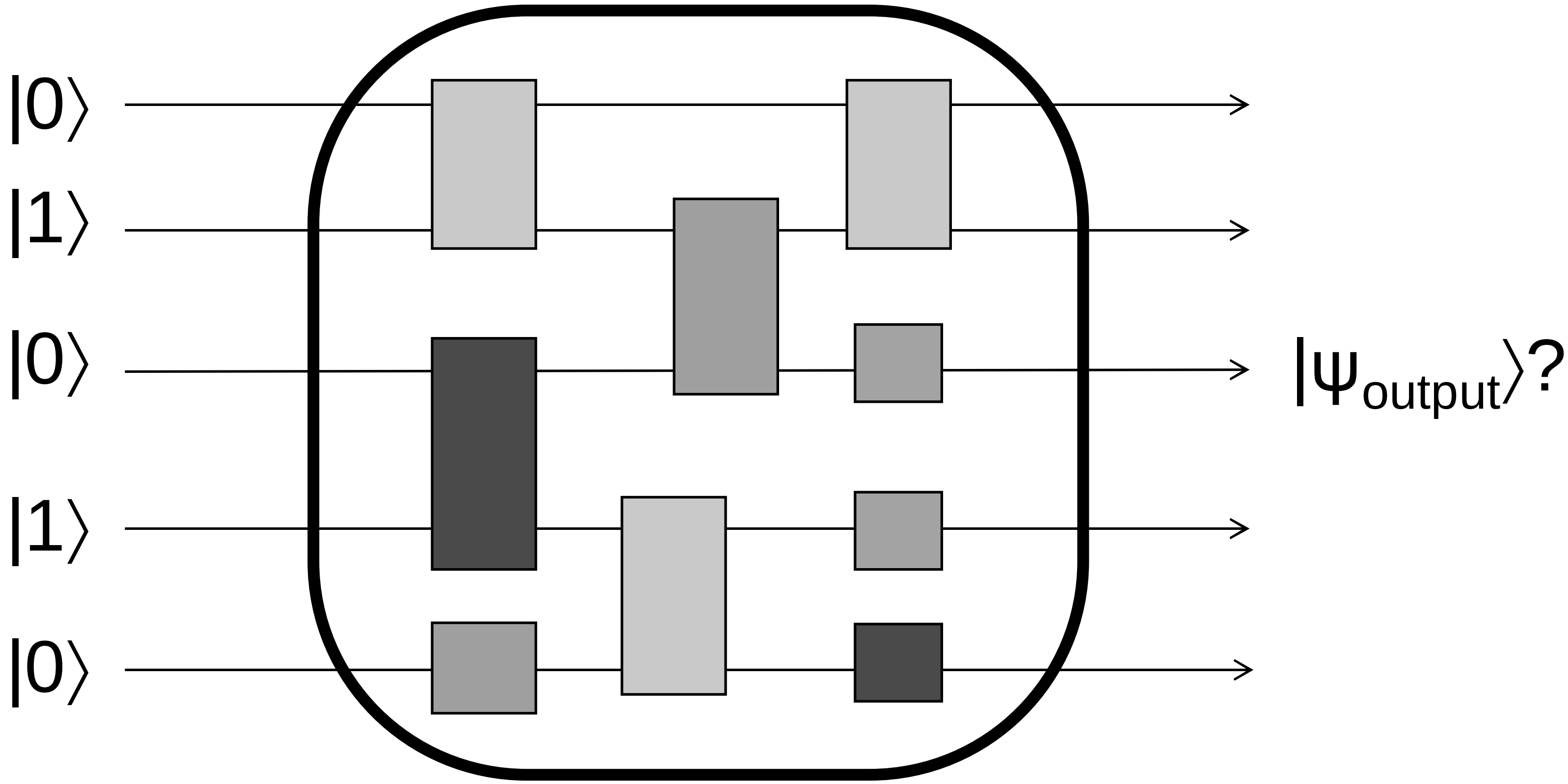
Engineering 1, Room 5109
vandam@cs

http://www.cs.ucsb.edu/~vandam/teaching/CS290/

Administrivia

- Exercises have been posted.
  Try to solve them,
  get help if you have problems
- Questions about the questions?
- Other questions?

Efficient Quantum Circuits



- Start with n classical bits as input.
- Apply a sequence of poly(n) elementary gates
- Measure the outcome ψ_{output}.

This Week

Mathematics of Quantum Mechanics:

- Braket calculus.
- Finite dimensional unitary transformations; eigenvector/eigenvalue decompositions.
- Projection Operators.

Circuit Model of Quantum Computation:

- Examples of important gates.
- Composing quantum gates into quantum circuits.
- (Classical) Reversible computation.
- Universality results for quantum circuits.

Hermitian Conjugates

- See handout “Mathematics of Quantum Computation”
- Generalization of complex conjugate* to matrices.
- Procedure: “Flip & conjugate”
- Notation: $|\psi\rangle^\dagger = \langle\psi|$ for vectors and $M^\dagger$ for matrices:

$$|\psi\rangle^\dagger = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_D \end{pmatrix}^\dagger = \begin{pmatrix} \overline{\alpha}_1 & \overline{\alpha}_2 & \cdots & \overline{\alpha}_D \end{pmatrix} = \langle\psi|$$

$$\begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1D} \\ M_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ M_{D1} & \cdots & \cdots & M_{DD} \end{pmatrix}^\dagger = \begin{pmatrix} \overline{M}_{11} & \overline{M}_{21} & \cdots & \overline{M}_{D1} \\ \overline{M}_{12} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \overline{M}_{1D} & \cdots & \cdots & \overline{M}_{DD} \end{pmatrix}$$

Inner / Outer Products

- $|x\rangle$ is a column vector, $\langle x|$ is a row vector.
- Inner Product $\langle x|y\rangle$ gives a $\mathbb{C}$-valued scalar
- Outer product $|y\rangle\langle x|$ gives a $D\times D$ $\mathbb{C}$-valued matrix:

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_D \end{pmatrix} \cdot \begin{pmatrix} \overline{\beta}_1 & \overline{\beta}_2 & \cdots & \overline{\beta}_D \end{pmatrix} = \begin{pmatrix} \alpha_1\overline{\beta}_1 & \alpha_1\overline{\beta}_2 & \cdots & \alpha_1\overline{\beta}_D \\ \alpha_2\overline{\beta}_1 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \alpha_D\overline{\beta}_1 & \cdots & \cdots & \alpha_D\overline{\beta}_D \end{pmatrix}$$

Notation: $|r\rangle\langle c|$ with $r,c\in\{1,\ldots,D\}$ denotes the 0-matrix, with a “1” in the r-th row and c-th column.

Hence for matrices $M = \sum_{ij} M_{ij}|i\rangle\langle j|$ and $M^\dagger = \sum_{ij} M^*_{ji}|i\rangle\langle j|$

Products of Bras and Kets

- How to deal with product sequences?
- Leave out the bars and dots: $\langle\psi|\cdot|\varphi\rangle = \langle\psi|\varphi\rangle$
- They don't commute: $\langle\varphi|\psi\rangle \neq \langle\psi|\varphi\rangle$
- Keep on eye on the dimensions:
  $|\psi\rangle$ is a vector, $\langle\psi|\psi\rangle$ a scalar and $|\psi\rangle\langle\psi|$ is a matrix.
- They are distributive and associative:
  $\langle\varphi|(\alpha|\psi\rangle+\beta|\psi'\rangle) = \alpha\langle\varphi|\psi\rangle+\beta\langle\varphi|\psi'\rangle$
  $(|\psi\rangle\langle\varphi|)(|\varphi\rangle\langle\psi|) = |\psi\rangle(\langle\varphi|\varphi\rangle)\langle\psi| = |\psi\rangle\langle\psi|$

Preserving Norms

- The norm of a vector $\alpha|v\rangle+\beta|w\rangle$, is determined by:
$\|\alpha|v\rangle+\beta|w\rangle\|^2 = (\alpha^*\langle v|+\beta^*\langle w|)(\alpha|v\rangle+\beta|w\rangle) =$
$\alpha^*\alpha\langle v|v\rangle + \beta^*\beta\langle w|w\rangle + \alpha^*\beta\langle v|w\rangle + \beta^*\alpha\langle w|v\rangle =$
$\alpha^*\alpha + \beta^*\beta + 2\mathrm{Real}(\alpha^*\beta\langle v|w\rangle)$

- Two vectors $|v\rangle$, $|w\rangle$ are **mutually orthogonal**, if and only if $\langle v|w\rangle = 0$; in which case $\|\alpha|v\rangle+\beta|w\rangle\|^2 = |\alpha|^2+|\beta|^2$.

- If T is a linear, norm preserving transformation of $|v\rangle$,$|w\rangle$, then the inner product between $(T|v\rangle)^\dagger$ and $T|w\rangle$ has to be the same as $\langle v|w\rangle$.
Hence: **T has to be inner product preserving.**

Unitarity 1

- Let M be a linear, norm preserving (= unitary) D-dimensional transformation on the Hilbert space $\mathbb{C}^D$.
- When represented as a $D \times D$ $\mathbb{C}$-valued matrix, how do we determine that M is unitary?
- Because $M|1\rangle$, $M|2\rangle, \ldots,$ $M|D\rangle$ have to have norm one, the columns of M have to have norm one.
- Because $|1\rangle$, $|2\rangle, \ldots,$ $|D\rangle$ are mutually orthogonal, the columns of M have to be mutually orthogonal.

Unitarity 2

- Let $M \in \mathbb{C}^{D \times D}$ be the matrix of a unitary transformation.
- The columns $M|1\rangle$, $M|2\rangle, \ldots, M|D\rangle$ have to form a D-dimensional orthonormal basis, hence $M^{\dagger} \cdot M = I$:

$$M^{\dagger} \cdot M = \begin{pmatrix} \longleftrightarrow \\ \longleftrightarrow \\ \longleftrightarrow \\ \\ \longleftrightarrow \end{pmatrix} \cdot \begin{pmatrix} \Big\updownarrow & \Big\updownarrow & \Big\updownarrow & & \Big\updownarrow \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} = I$$

- M is invertible: $M^{-1} = M^{\dagger}$, which is also unitary.
- The identity matrix is unitary
- The set of D-dimensional unitary transformations is a (matrix) group.

Recognizing Unitarity

- Perform the matrix multiplication: $M^{\dagger} \cdot M = M \cdot M^{\dagger} = I$? Simple for small matrices, impractical for larger ones.

- Prove that $M|1\rangle, \ldots, M|D\rangle$ are mutually orthogonal.

- If M is a classical computation, then the above means that $M|1\rangle, \ldots, M|D\rangle$ has to be a permutation. Alternatively, a classical M has to be reversible.

- Topic of (classical) reversible computation.

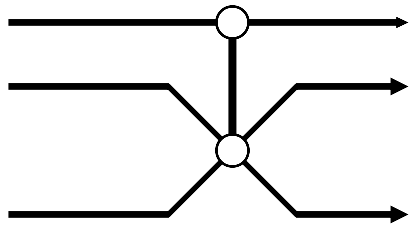
Reversible Computation

- Standard computation is irreversible: (a,b) $\mapsto$ (a AND b)

- Reversible gates have FAN-IN = FAN-OUT.

- Irreversible gates: (a,b) $\mapsto$ (a OR b), (a) $\mapsto$ (0),
  but also: (a,b) $\mapsto$ (a, a OR b)

- Reversible gates: (a) $\mapsto$ (~a), CNOT:(a,b) $\mapsto$ (a, b$\oplus$a),
  CCNOT:(a,b,c) $\mapsto$ (a,b,c$\oplus$ab), and C-SWAP:



$$\text{C-SWAP}: |0,b,c\rangle \mapsto |0,b,c\rangle$$
$$\text{C-SWAP}: |1,b,c\rangle \mapsto |1,c,b\rangle$$

Reversibility Issues

$$|x\rangle \longrightarrow \boxed{F} \longrightarrow |F(x)\rangle$$

For general $F:\{0,1\}^n \to \{0,1\}^n$
$|x\rangle \mapsto |F(x)\rangle$ is irreversible

$$|x\rangle \longrightarrow \boxed{F} \longrightarrow |F(x)\rangle$$

For reversible $F:\{0,1\}^n \to \{0,1\}^n$
$|x\rangle \mapsto |F(x)\rangle$ is reversible

$$|x,y\rangle \longrightarrow \boxed{Id,\oplus F} \longrightarrow |x,y\oplus F(x)\rangle$$

For general $F:\{0,1\}^n \to \{0,1\}^n$
$|x,y\rangle \mapsto |x,y\oplus F(x)\rangle$ is reversible

Which reversible functions can we implement efficiently under the assumption that we can implement F efficiently?

CC-NOTs as Universal Gates

- With CCNot gates, we can implement NOT and AND: CCNOT:$|1,1,c\rangle \mapsto |1,1,\sim c\rangle$, CCNOT:$|a,b,0\rangle \mapsto |a,b,ab\rangle$.
- If we keep old memory around, any circuit function F can be implemented efficiently $|x,0,0\rangle \mapsto |x,g_x,F(x)\rangle$
- By copying the output F(x) and running the circuit in reverse, we can erase the garbage bits g_x: $|x,g_x,F(x),0\rangle \mapsto |x,g_x,F(x),F(x)\rangle \mapsto |x,0,0,F(x)\rangle$.
- In sum: $|x,0,0\rangle \mapsto |x,F(x),0\rangle$ can be implemented efficiently as long as we have clean 0-qubits around.

Power of Reversible Computation

- We showed that the requirement of reversibility does not change (significantly) the efficiency of our computations: **Reversible Computation = General Computation.**
- But what about the efficiency of implementing of other reversible computations?

Problematic Reversibility

- If F is a reversible function (a permutation of $\{0,1\}^n$), then $|x\rangle \mapsto |F(x)\rangle$ is reversible.

- Even if F can be implemented efficiently (classically), it does not always hold that $|x\rangle \mapsto |F(x)\rangle$ can be implemented in a unitary/reversible way.

- $|x,0\rangle \mapsto |x,F(x)\rangle$ can be done efficiently, but $|x,F(x)\rangle \mapsto |0,F(x)\rangle$ can be hard.
- Reason: F^{-1} may be hard to implement (one-way F).

More on Reversibility

- Reversibility also plays a role in the heat production of bit operations: $k_B T \ln(2) \sim 10^{-22}$ Joule per bit.
- Remember: A Quantum Computation can always just as easily be done in reverse:
  Just read the circuit right from left,
  and invert each unitary gate along the way.
- See in “Quantum Computation and Quantum Information”: §3.2.5, “Energy and Computation”