Ancilla factories

Universal distillery

Data

Encoded linear gates for teleportation

Encoded noisy "magic" states

Encoded nonlinear gates for teleportation

Ancilla factories

Universal distillery

Ben Reichardt
Caltech
Motivations for quantum information processing

- Quantum computing (QC)
  - Extended Church-Turing Thesis: Anything physically efficiently computable can be computed efficiently on my laptop
  - QC: Extended Church-Turing Thesis is false; there are exponentially-faster algorithms (for interesting problems) by using quantum mechanics
- Cryptography
  - Breaks RSA public-key cryptosystem
  - Gives unconditionally secure key distribution
- Simulation & modeling
  - for quantum devices,
  - chemistry,
  - materials (high-T superconductors, new states of matter?)
- Quantum sensing
  - Precise measurement and lithography
  - Atomic clocks
- Basic science
  - Investigate measurement/decoherence, quantum/classical boundary
  - Test qu. mechanics on new scales

(but no free lunch…)
Quantum information

- “Qubit”:
  \[
  \left( \begin{array}{c}
  \alpha_0 \\
  \alpha_1
  \end{array} \right) = \alpha_0 |0\rangle + \alpha_1 |1\rangle
  \]

- State of \( n \) qubits = unit vector in \( \mathbb{C}^{2^n} \)

\[
|\alpha_0|^2 + |\alpha_1|^2 = 1
\]

\[
(\alpha_x)_{x \in \{0,1\}^n}
\]
Quantum information

- “Qubit”: \( \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \alpha_0 |0\rangle + \alpha_1 |1\rangle \)
  \[ |\alpha_0|^2 + |\alpha_1|^2 = 1 \]

- State of \( n \) qubits = unit vector in \( \mathbb{C}^{2^n} \)

- Computation by local gates, rotate the state vector

- Observing/measuring system collapses it to a single classical bitstring \( x \)
  - No exponential parallelism
  - Have to “finesse” the quantum system to output the classical information you want
Classical information processing

• Classical state is a vector of probabilities:

\[ \{ p_x \}_{x \in \{0,1\}^n} \quad p_x \geq 0 \quad \sum_x p_x = 1 \]

• Valid operations are stochastic maps

Quantum information processing

• Quantum state is also a vector

\[ \{ \alpha_x \}_{x \in \{0,1\}^n} \quad \sum_x |\alpha_x|^2 = 1 \]

• Valid operations are rotations (unitaries)

The universe is quantum mechanical but it looks classical because of noise…
Quantum algorithms

Simulation
- ...of dynamics of physical quantum systems
- Approx. Jones polynomial

Search & random walk
- Grover search
- Element distinctness
- Graph traversal

Factoring
- Discrete log

Fourier sampling & Hidden subgroup
- Abelian, some nonabelian HSPs
- Pell's equation
- Graph isomorphism???

Today: New algorithmic approach based on span programs

Exponential speedups

Polynomial speedups

CS+Physics

Exponential speedups

Polynomial speedups

CS
Quantum computing in 2008

- **Ion traps**
  - can trap and cool 16-18 qubits
  - can entangle 6-8 qubits in a trap
  - microfabrication of trap arrays on chips, dealing with increased noise
  - in next 2-3 years may be able to compute with 40-60 qubits
  - challenges: controlling thousands of traps with dozens of detection channels and lasers along the surface of the chip…

- **Superconducting qubits**
  - 2 qubit local interactions becoming routine
  - nonlocal movement & interactions now possible
  - noise levels seem promising…

- **Other technologies:**
  - Photonic qubits, quantum dots…

---

[Image: nature](https://www.nature.com)
• Scaling these systems is a major engineering challenge

• But the basic technologies have been proven, there are intermediate rewards

• And there are no known fundamental difficulties, except…


Common obstacle is noise!

• Physically reasonable noise rates are ~1% error per gate, or maybe 0.1%
  \[ \therefore \text{Only 100 operations before an error can occur and propagate through the system} \]

• Factoring a 2048-bit number uses
  • \( 6 \times 10^{11} \) gates on
  • 10,000 qubits
  • Need error \( \lesssim 1/10^{12} \) per gate

\[ \text{K-bit number:} \]
\[ 72 K^3 \text{ gates} \quad \text{versus} \quad e^{K^{1/3}} \text{classically} \]
\[ 5 K \text{ qubits} \]
Noise is fundamental problem for quantum computers: entangled systems are fragile

- Schrödinger’s cat:

  \[
  \left( \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right) \frac{1}{\sqrt{2}} (|\text{live cat}\rangle + |\text{dead cat}\rangle)
  \]

  i.e.

- “Both dead and alive,” in superposition; but collapses to one or the other when observed.

- A single stray photon can collapse it — and also analogous states in a quantum computer.

- Physically reasonable noise rates are ~1% error per gate, or perhaps 0.1%.
How to deal with noise?

1. Engineering
   - Not enough—noise is fundamental in quantum systems

2. Fault tolerance
   - Enough to engineer the noise rate beneath a constant threshold,
   - Then effective noise rate can be decreased arbitrarily (and efficiently) using error-correcting codes

[Von Neumann ’56]
**Classical fault tolerance**

[Von Neumann '56]

Make fault-tolerant a circuit consisting of a universal set of operations, some faulty:

![Diagram of fault tolerance circuit]

Perfect op's: \[\begin{cases} 0, \\ 1, \end{cases}\]

Faulty op's: \{AND, NOT\}

Encoding:

\[
\begin{array}{c}
0_L = \\
1_L = \\
\end{array}
\begin{array}{c}
0 \\
0 \\
0 \\
1 \\
1 \\
1 \\
\end{array}
\]

Transversal gate application:

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Error correction:

![Graph showing error correction]
What’s different quantumly?

- Quantum problems:
  - Quantum states are continuous, not discrete—need to protect against continuous errors
  - No-cloning theorem: Can’t copy a quantum state $|\psi\rangle \leftrightarrow |\psi\rangle|\psi\rangle$, so no immediate analog of the repetition code $0 \leftrightarrow 0^n$, $1 \leftrightarrow 1^n$

- But quantum ECCs do exist! [Shor ’95]
Operational def. of QECC

• Quantum problems:
  • Quantum states are continuous, not discrete—need to protect against continuous errors
  • No-cloning theorem: Can’t copy a quantum state $|\psi\rangle \mapsto |\psi\rangle|\psi\rangle$, so no immediate analog of the repetition code $0 \mapsto 0^n$, $1 \mapsto 1^n$

• But quantum ECCs do exist! [Shor ’95] Operationally,

\[
|\psi\rangle \xrightarrow{\text{encode}} E(|\psi\rangle) \xrightarrow{\text{noise}} N^\otimes m(E(|\psi\rangle)) \xrightarrow{\text{recover}} E(|\psi\rangle)
\]
Quantum error-correcting codes exist

- Although quantum states are continuous, correcting a *discrete* set of errors (bit and phase flips) suffices
- Based on classical linear ECCs: QECC comes from two linear ECCs (one for bit flips, one for phase flips)
Quantum error-correcting codes exist

- Although quantum states are continuous, correcting a *discrete* set of errors (bit and phase flips) suffices
- Based on classical linear ECCs: QECC comes from *two* linear ECCs
- How can we *use* these codes?
  - Need operations as well as memory
  - Error recovery must be resilient to faults during recovery
  - How to encode into them in the first place?? (qu problem)

\[
\begin{align*}
|\psi\rangle & \xrightarrow{\text{encode}} \mathcal{E}(|\psi\rangle) \\
\text{data qubits} & \quad \text{encoded data} \\
\mathcal{N}^\otimes m(\mathcal{E}(|\psi\rangle)) & \quad \text{noise} \quad \text{recover} \\
\text{encoded data} & \quad \mathcal{E}(|\psi\rangle) \\
\end{align*}
\]
Fault-tolerance intuition

- **Compile** ideal circuit into “fault-tolerant” (noise-resistant) version, starting with small QECC:

  ![Diagram](image)

  - **Gate**
  - **Encoded gate**
  - **Error correction**
  - **Perfect decoding**
  - **Perfect gate**

  Prob. diagram fails to commute

  Threshold for improvement: $1/c$

- **Concatenate** (i.e., repeat) for arbitrarily improved reliability (so arbly long calcs), if starting below a constant noise threshold

- **Problem**: Noise model at encoded level is not the same as the physical noise model!
Abridged History of Quantum Fault Tolerance

- 1996-97: First fault-tolerance results: QECCs, threshold proofs
  Shor, Steane, Calderbank, Aharonov, Ben-Or, Kitaev, Knill,
  Laflamme, Zurek, …
- Proved existence of some positive tolerable noise rate using
  concatenated qu. codes of distance $\geq 5$
- No explicit lower bounds on tolerable noise rate, but
  estimates were $10^{-6}-10^{-5}$ noise per gate
- Moral: Fault tolerance makes quantum computers plausible
  in the real world

"Dark Ages"
-D. Gottesman
Abridged History of Quantum Fault Tolerance

Proofs

- 1997: Aharonov/Ben-Or, Kitaev: Prove positive tolerable noise rate for codes of distance $d \geq 5$
- 2005: R, Aliferis/Gottesman/Preskill: First explicit numerical threshold lower bounds, threshold for distance-3 codes

Estimates & simulations

- 2002: Steane: Correct bit flip errors all at once, and then phase flip errors all at once
  - based on simulations, estimates $3 \times 10^{-3}$ tolerable noise rate per gate

Simulations using distance-3 codes

- Basic estimates:
  - Aharonov & Ben-Or ‘97
  - Gottesman ‘97
  - Knill-Laflamme-Zurek ‘98
  - Preskill ‘98
- Optimized estimates:
  - Zalka ‘97
  - R ‘04
  - Svore-Cross-Chuang-Aho ‘05
- 2D locality constraint
  - Szkopek et al ‘04
  - Svore-Terhal-DiVincenzo ‘05
Improved threshold result \cite{R04}:
- Modification of standard error correction scheme increases estimated threshold 3x, to almost 1%.

**Abridged History of Quantum Fault Tolerance**

**Proofs**

- 1997: Aharonov/Ben-Or, Kitaev: Prove positive tolerable noise rate for codes of distance $d \geq 5$
- 2005: R, Aliferis/Gottesman/Preskill: First explicit numerical threshold lower bounds, threshold for distance-3 codes

**Estimates & simulations**

- 2002: Steane: Correct bit flip errors all at once, and then phase flip errors all at once
  - based on simulations, estimates $3 \times 10^{-3}$ tolerable noise rate per gate

**Postselection**
error correction detection!
Error-detection-based fault-tolerance

- **Compile** ideal circuit into “fault-tolerant” (noise-resistant) version, starting with small QECC:

  ![Diagram of error-detection-based fault-tolerance](image)

  - Prob. diagram fails to commute
  - Threshold for improvement: $1/c$
  - Physical error rate $p$

  - In simulations, tolerates much higher noise rates than error-correction-based FT schemes
  - But (previously) no proven positive threshold!
Effect of postselection in ancilla preparation

[R '04]
Abridged History of Quantum Fault Tolerance

Proofs

• 1997: Aharonov/Ben-Or, Kitaev: Prove positive tolerable noise rate for codes of distance \( d \geq 5 \)

• 2005: R, Aliferis/Gottesman/Preskill: First explicit numerical threshold lower bounds, threshold for distance-3 codes

Estimates & simulations

• 2002: Steane: Correct bit flip errors all at once, and then phase flip errors all at once
  • based on simulations, estimates \( 3 \times 10^{-3} \) tolerable noise rate per gate

• Postselection

• Postselection + Teleportation

Improved threshold result \([R \ '04]\)
  • Modification of standard error correction scheme increases estimated threshold 3x, to almost 1%.

Knill’s threshold result \([Knill \ '04]\)
  • Estimated 3-6% threshold for independent depolarizing errors.
Quantum teleportation

Alice’s lab

Bob’s lab

prepare entangled state

measure

classical measurement outcome

time

$|\psi\rangle$

$|\psi\rangle$
Applying teleportation to fault tolerance


\[ |\psi\rangle \]

prepare entangled state

\[ |\psi\rangle \]

measure

\[ |\psi\rangle \]

time
Applying teleportation to fault tolerance

1. Error correction
2. Computation

* decoding measurements using classical computer
Applying teleportation to fault tolerance

1. Error correction  2. Computation

$|\psi\rangle$  \hspace{1cm} $U |\psi\rangle$

* operation $U$ should be “$C_2$”
Applying teleportation to fault tolerance
Error correction + Encoded Computation

- Teleportation allows for correcting bit flip errors, phase flip errors, and doing one step of computation all at once.
- (Provided that we can prepare reliably the necessary resource states.)
Teleportation allows for correcting bit flip errors, phase flip errors, and doing one step of computation all at once.

(Provided that we can prepare reliably the necessary resource states.)

**Note:** We can prepare very good ancilla states, e.g., throwing away all ancillas with any detected errors (“postselection”). We wouldn’t want to throw away the data—but the data is isolated from the ancilla state.
Teleportation allows for correcting bit flip errors, phase flip errors, and doing one step of computation all at once.

(Provided that we can prepare reliably the necessary resource states.)

**Note:** We can prepare very good ancilla states, e.g., throwing away all ancillas with any detected errors (“postselection”). We wouldn’t want to throw away the data—but the data is isolated from the ancilla state.

### Quantum disadvantages
- States are continuous (i.e., analog)
- No-cloning theorem

### Quantum advantage!
- Quantum teleportation allows isolating the data from errors
• **Problem:** Although Knill *estimated* tolerable noise rate was 3-6%, proofs could not show that postselection-based schemes tolerated any noise at all!

**Renormalization frustrates previous proofs**

Most of the time, errors are detected — but (counterintuitively) survival probability for uncontrolled portion could be much *higher*.

Proofs based on controlling events most of the time, with occasional failures.

Uncontrolled fraction of probability mass increases exponentially after renormalizing!
**Intuition for Aharonov/Ben-Or’s proof**

- **Idea:** Maintain inductive invariant of goodness. (A level-k block is good “if it has at most one bad level-(k-1) subblock.”)

  ![Diagram of good and bad states](image)

  (assuming one level k-1 error, m≥7)

- **Problems:**
  - Inefficient analysis: Logical error rate for a distance-d code drops as $c \cdot p^{(d-1)/2}$ instead of $c \cdot p^{(d+1)/2}$.
    - Can’t hope for very good rigorous lower bounds on the noise threshold
  - No threshold at all for concatenated d=3 codes, or for postselection-based schemes

  ![Diagram of good and bad states](image)

  (one level k-1 error is already too many)
• **Problem:** Although Knill *estimated* tolerable noise rate was 3-6%, proofs could not show that postselection-based schemes tolerated any noise at all!

• Renormalizing the error distribution leads to bad correlations.

**Results**  
[R ’06]

• *Existence* of tolerable noise rates for many fault-tolerance schemes, including:
  
  • Schemes based on error-*detecting* codes, not just ECCs (Knill-type)
  
  • Distance-3 codes, and more efficient “Fibonacci”-type schemes (d=2 codes)

• Tolerable threshold *lower bounds* *

  • 0.1% simultaneous depolarization noise†

  • 1.1%, if error model known exactly

* Subject to minor numerical caveats  
† Versus .02% best lower bound for error-correction-based FT scheme [Aliferis, Cross 2006]
Techniques

- Main new technique is to maintain close control over the distribution of errors in the quantum computer (Previous threshold proofs had used a “worst-case” criterion for error behavior that blew up during renormalization.)
- Rewrite true error distribution as a mixture of nearby distributions whose error distributions lack nasty correlations.
Bitwise-independent noise is nice...

- **Def:** Noisy encoder = perfect encoder, followed by bitwise-indep. noise at rates $\leq p$.

\[ \tilde{E} = E \]

(tool for analysis—such encoders don’t actually exist)

**Induction claim?** (much stronger than Aharonov/Ben-Or’s claim)

- bitwise-independent errors preceding encoded gate
- bitwise-independent errors following perfect gate, plus quadratically suppressed independent logical errors
Since the error model is preserved (level-one logical errors have the same form as physical errors), the analysis can be repeated to give a threshold
Bitwise-independent noise is nice...

- **Def:** Noisy encoder = perfect encoder, followed by bitwise-indep. noise at rates \( \leq p \).

\[ \tilde{\mathcal{E}} = \mathcal{E} \]

(tool for analysis—such encoders don’t actually exist)

**Induction claim**

- bitwise-independent errors preceding encoded gate
- bitwise-independent errors following perfect gate, plus quadratically suppressed independent logical errors

(much stronger than Aharonov/Ben-Or's claim)
Details in proving that mixing works

- **Numerical** approach (for numerical threshold lower bounds)

- **Existence** argument (for threshold existence proofs):
  - characterize convex hull of dit-wise independent distributions (a simplex)
  - “pull back” actual distribution onto distribution on dits

- Must also obtain **universality** — CNOT and similar “linear” gates can be efficiently simulated on a classical computer. Need a nonlinear operation (AND or Toffoli). Use “magic states distillation.”
Conclusions

• Conclusion: Mixing argument shows that concatenation works to reduce errors. Error events are correlated, but error correlations do not explode.

• Correlations manifest themselves as asymmetries in the conditional error models

—violates a key assumption of Knill, that all gates have symmetrical failure models, at all levels of concatenation

• With postselection, gate error rates that are asymmetrically too low can be just as bad as error rates that are too high

• Are Knill’s simulations too optimistic?
More fault-tolerance work...

- Magic states distillation (for teleporting into a universal gate set) [R’05, ‘06]
- Optimizing ancilla verification [R’06]
- Higher-order-accurate composite pulses, based on quantum search algorithm, for eliminating noise *without encoding* [R’05]
Open questions in fault tolerance

• Quantum computers need fault-tolerance techniques if they are to scale, but…

• Current FT schemes are not good enough
  • Need to increase tolerable noise rate, reduce overhead
  • So far, the biggest improvements have come not from optimizations or customizations, but rather from new quantum concepts that unify. The next division to remove is the code concatenation levels.

• Foundations:
  • Extend applicability of threshold proofs
  • Improve threshold upper bounds

• Connecting full-blown fault-tolerance schemes to implementations
  • Specialized, low-level error prevention (e.g., composite pulses, DFSs)
Scatter a wave against the tree...
FGG quantum walk $|\psi_t\rangle = e^{iAGt}|\psi_0\rangle$
FGG quantum walk $|\psi_t\rangle = e^{iAGt}|\psi_0\rangle$
FGG quantum walk \[ |\psi_t\rangle = e^{iA_G t} |\psi_0\rangle \]

\[ x_{11} = 1 \]  
Wave reflects!

\[ x_{11} = 0 \]  
Wave transmits!
Farhi, Goldstone, Gutmann ‘07 algorithm

• **Theorem** ([FGG ‘07, CCJY ‘07]): A balanced binary NAND formula can be evaluated in time $N^{1/2+o(1)}$.

**Questions:**

1. Why does it work?

2. How does it connect to what we know already?

3. How does it generalize?

4. What kinds of problems can we hope to solve with this technique?
**Farhi, Goldstone, Gutmann ‘07 algorithm**

- **Theorem** ([FGG ‘07, CCJY ‘07]): A balanced binary NAND formula can be evaluated in time $N^{1/2+o(1)}$.

**Questions:**

1. Why does it work?

2. How does it connect to what we know already?

3. How does it generalize?

4. What kinds of problems can we hope to solve with this technique?

**Answers:**

- “span programs” [Karchwer/Wig. ‘93]
- formula evaluation problem over extended gate sets
Farhi, Goldstone, Gutmann ‘07 algorithm

- **Theorem** ([FGG ‘07, CCJY ‘07]): A balanced binary NAND formula can be evaluated in time $N^{1/2+o(1)}$.

**Questions:**

1. Why does it work?

2. How does it connect to what we know already?

3. How does it generalize?

4. What kinds of problems can we hope to solve with this technique?

5. No, really, **WHY** does it work?

**Answers:**

- “span programs” [Karchwer/Wig. ‘93]

- formula evaluation problem over extended gate sets

- ???
**[FGG ‘07] algorithm**

- **Theorem** ([FGG ‘07, CCJY ‘07]): A balanced binary AND-OR formula can be evaluated in time $N^{1/2+o(1)}$.

  Analysis by scattering theory.

**[ACRŠZ ‘07] algorithm**

- **Theorem**:
  - An “approximately balanced” AND-OR formula can be evaluated with $O(\sqrt{N})$ queries (optimal!).
  - A general AND-OR formula can be evaluated with $N^{1/2+o(1)}$ queries.

Running time is $N^{1/2+o(1)}$ in each case, after preprocessing.

**[RŠ ‘08] algorithm**

- **Theorem**: A balanced (“adversary-bound-balanced”) formula $\varphi$ over a gate set including all three-bit gates (and more…) can be evaluated in $O(\text{ADV}(\varphi))$ queries (optimal!).

  (Some gates, e.g., AND, OR, PARITY, can be unbalanced—but not most!)
Recursive 3-bit majority tree

- Best quantum lower bound is
  \[ \Omega(\text{ADV}(\varphi) = 2^d) \]  
  [LLS’05]

- Expand majority into \{\text{AND, OR}\} gates:
  \[
  \text{MAJ}_3(x_1, x_2, x_3) = (x_1 \land x_2) \lor (x_3 \land (x_1 \lor x_2))
  \]

  \[ \therefore \{\text{AND, OR}\} \text{ formula size is } \leq 5^d \]

  \[ \therefore O(\sqrt{5^d}) = O(2.24^d)-\text{query algorithm} \]

  [ACRSZ ‘07]

[RŠ ‘08] algorithm

- **Theorem:** A balanced (“adversary-bound-balanced”) formula \(\varphi\) over a gate set including all three-bit gates (and more…) can be evaluated in \(O(\text{ADV}(\varphi))\) queries (optimal!).

  (Some gates, e.g., AND, OR, PARITY, can be unbalanced—but not most!)

- New: \(O(2^d)\)-query quantum algorithm
Span program definition

- **Def:** An n-bit span program $P$ is:
  - A target vector $t$ in vector space $V$ over $\mathbb{C}$,
  - $n$ input subspaces, one for each bit

Span program $P$ computes $f_P: \{0,1\}^n \rightarrow \{0,1\}$,

$$f_P(x) = 1 \iff t \text{ lies in the span of } \{ \text{subspace } i : x_i=1 \}$$

- **Ex.:** $P$:

```
x_3 = 1
x_2 = 1
x_1 = 1
```

$\Rightarrow f_P = \text{MAJ}_3$

*Not the general def.*
Span program $P$

E.g., $\text{MAJ}_3$:

\[
\begin{align*}
&x_1 = 1 \\
&x_2 = 1 \\
&x_3 = 1
\end{align*}
\]

Matrix

\[
t = \begin{pmatrix}
1 \\
0
\end{pmatrix} \begin{pmatrix}
1 & 0 & -1 \\
1 & 1 & 1 \\
x_1 &= 1 \\
x_2 &= 1 \\
x_3 &= 1
\end{pmatrix}
\]

For a given $x$, add edges above those inputs evaluating to false.

Weighted bipartite graph

output edge
Span program $P$

E.g., $\text{MAj}_3$:

$$t = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Matrix

$$t = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 1 & 1 \end{pmatrix}$$

$\lambda=0$ eigenvector computes $P$

For a given $x$, add edges above those inputs evaluating to false.

**Thm:** $f_P(x) = 1 \leftrightarrow \exists$ eigenvalue-0 eigenvector supported on bottom vertex.

Weighted bipartite graph
Recursive MAJ\(_3\)

**Main Theorem:**

- \( \varphi(x) = 1 \Rightarrow A_{G(x)} \) has \( \lambda = 0 \) eigenstate with \( \Omega(1) \) support on the root.
- \( \varphi(x) = 0 \Rightarrow A_{G(x)} \) has no eigenvectors overlapping the root with \( |\lambda| < 1/2^d \).

\( \Rightarrow O(2^d) \)-query (optimal!) recursive MAJ\(_3\) evaluation algorithm
<table>
<thead>
<tr>
<th>Problem</th>
<th>Classical</th>
<th>Quantum</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 \land x_2 \land \cdots \land x_N \lor \cdots \lor x_N$</td>
<td>$\Theta(N)$</td>
<td>$\Theta(\sqrt{N})$ [Grover '96]</td>
</tr>
<tr>
<td><strong>NP</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balanced AND-OR</td>
<td>$\Theta(N^{0.753\ldots})$ (fan-in two)</td>
<td>$\Theta(\sqrt{N})$ [FGG, ACRŠZ '07]</td>
</tr>
<tr>
<td></td>
<td>[S'85, SW'86, S'95]</td>
<td></td>
</tr>
<tr>
<td><strong>PSPACE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General read-once AND-OR</td>
<td>$\Omega(N^{0.51})$ [HW'91]</td>
<td>$\Omega(\sqrt{N})$, $\sqrt{N} \cdot 2^{O(\sqrt{(\log N)})}$ [BS '04] [ACRŠZ '07]</td>
</tr>
<tr>
<td></td>
<td>Conj.: $\Omega(D(f)^{0.753\ldots})$ [SW '86]</td>
<td></td>
</tr>
<tr>
<td>Balanced $\text{MAJ}_3$</td>
<td>$\Omega((7/3)^d), O((2.6537\ldots)^d)$ [JKS '03]</td>
<td>$\Theta(2^d=2^\log_3 2)$ and much more… [RŠ '08]</td>
</tr>
</tbody>
</table>
Open:

- Extensions to larger gate sets…
- Unbalanced formulas over more gates…
- Why do span programs work so well? Connection to adversary lower bounds $\text{ADV}(f) \leq \text{ADV}^\pm(f)$?

**Open ?: More quantum algorithms based on span programs?**

- Our quantum algorithm evaluates span programs. We’ve applied it by building a large span program by composing small ones for all the gates.
- New framework for developing quantum algorithms: Are there interesting quantum algorithms based directly on large span programs? (E.g., graph problems, Perfect Matching, …) [notion of quantum recursion]
Bird’s-eye view of quantum computing

What to do with a quantum computer?
- Adiabatic optimization alg.
- Formula evaluation

How can we build a quantum computer?
- Composite pulses
- Fault tolerance theory