Computational complexity: Space complexity, probabilistic MTs & other models

Petr Kurapov

MIPT

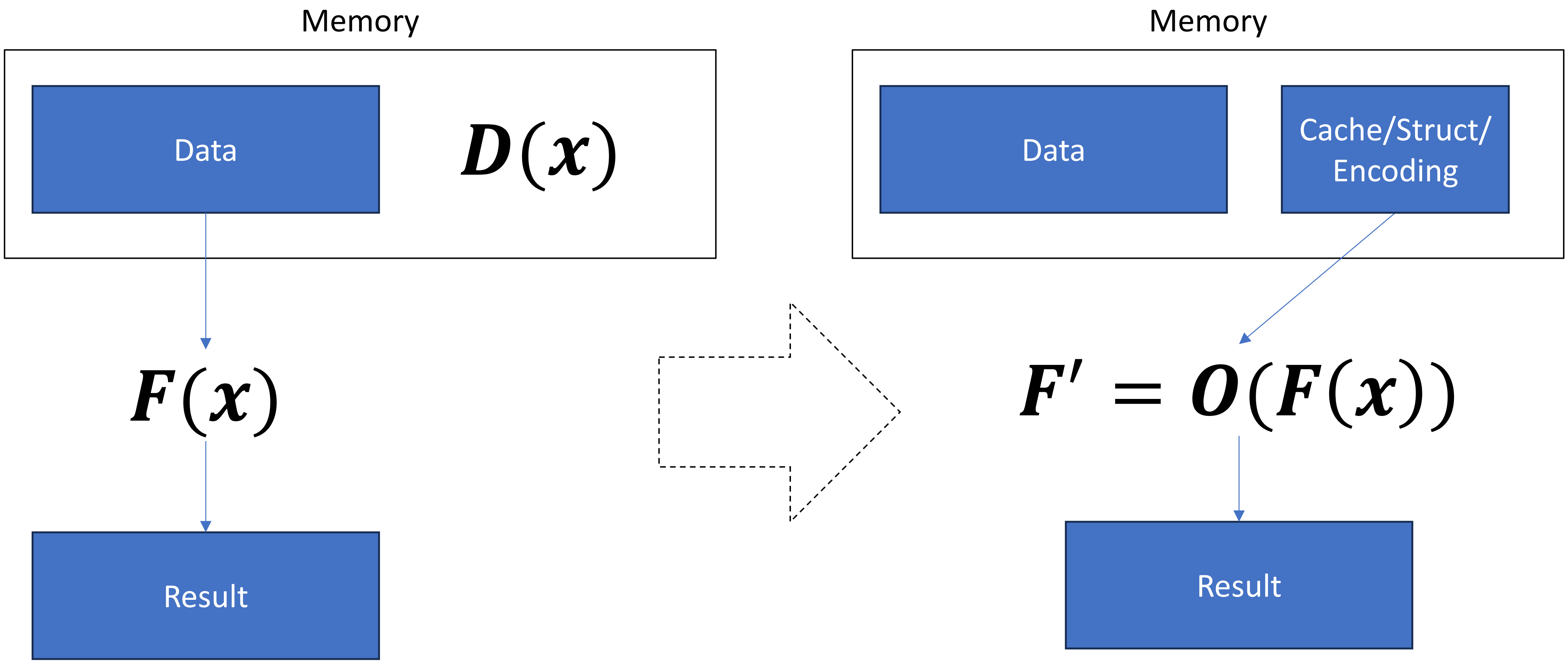
Fall 2024

Previous results

- Math model for computations – Turing machine (TM)
- There's a universal TM that can simulate any other efficiently
- Some functions are not computable by any TM
- Defined class of "easy" problems P (can be solved efficiently)
- NP class – verifiable in polynomial time.
- NP-completeness & NP-hardness
- 3SAT NP-completeness, Cook's theorem
- coNP, EXP, NEXP classes

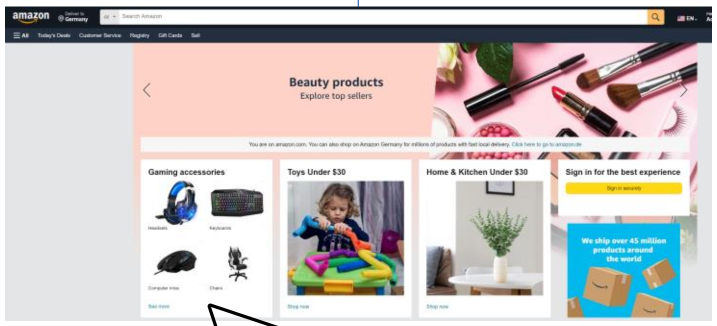
Space/Time tradeoff

$$D' = \Omega(D(x))$$



Space/Time tradeoff

```
INSERT INTO amazon_table (user_id,
product_id, timestep) VALUES (…);
```



Archive/Warehouse

Compression

```
SELECT records FROM amazon_table
WHERE timestep < now + 1h;
```

Space complexity

- Space bounded computation: S: $\mathbb{N}\rightarrow\mathbb{N}$, L $\subseteq \{0,1\}^* \rightarrow L \in (N)PSPACE\big(s(n)\big)\ if\ \exists c, (N)M: (N)M\ -$ $decides\ L\ using\ cs(n)\ tape\ cells$
- Restrict space bounds to space-constructible functions (There's a TM that calculates $S(|x|)\ in\ O(S|x|)$)
- Note: sub-linear space complexity does make sense (as opposed to time complexity), require at least $logn$ to store input indexes ($S(n) \ll n\ problem$?)
- $DTIME(S(n)) \subseteq PSPACE(S(n)) \subseteq NSPACE(S(n)) \subseteq DTIME(2^{O(S(n))})$

MT with **s** cells can at least run $2^{|x|}$ operations

n

Input tape	>	0	1	1	0	1	0	1		

$S(n)$

Step 0	0	0	0	0	0	0	0	0		
Step 1	0	0	0	0	0	0	0	1		
Step 2…	0	0	0	0	0	0	1	0		

…

$2^{|x|}$

Output tape	>	0	1	1	0	1	0	1		

Space complexity

- $DTIME(S(n)) \subseteq PSPACE(S(n)) \subseteq NSPACE(S(n)) \subseteq DTIME(2^{O(S(n))})$
- Configuration c_t - M state description: cursors positions, registers' states, work tape state, …

Configurations are unique!

Space complexity

- $DTIME(S(n)) \subseteq PSPACE(S(n)) \subseteq NSPACE(S(n)) \subseteq DTIME(2^{O(S(n))})$
- Configuration c_t - M state description
- $O(1) \cdot n \cdot |\Gamma|^{S(n)} = 2^{O\left(S(n)\right)+\log n} = 2^{O\left(S(n)\right)}$

Space complexity

- P & NP space analogies:
  - $\boldsymbol{PS} = \bigcup_{c>0} PSPACE(n^c)$
  - $\boldsymbol{NPS} = \bigcup_{c>0} NSPACE(n^c)$
- Sublinear classes:
  - $\boldsymbol{L} = PSPACE(\log n)$
  - $\boldsymbol{NL} = NSPACE(\log n)$
- 3SAT?

Space complexity

- P & NP space analogies:
  - $\boldsymbol{PS} = \bigcup_{c>0} PSPACE(n^c)$
  - $\boldsymbol{NPS} = \bigcup_{c>0} NSPACE(n^c)$
- Sublinear classes:
  - $\boldsymbol{L} = PSPACE(\log n)$
  - $\boldsymbol{NL} = NSPACE(\log n)$
- 3SAT?
- $NP \subseteq PS$ – iterate through all 2^k possible values using $O(n)$ memory

PS-completeness

- Same as NP completeness:
- L' – **PS**–hard if $L \leq_p L'$ for $\forall L \in \boldsymbol{PS}$
- L' - **PS**-complete if it's **PS**-hard & in **PS**

PS-completeness: example

- Quantified Boolean formula (QBF), prenex form:
- $z \in \{0,1\}^n$; $Q_i \in \{\forall, \exists\}$; $\varphi(z) = Q_1 z_1 \dots Q_n z_n f(z)$
- Ie:
  - $\forall x \exists y (x \wedge y) \vee (\bar{x} \wedge \bar{y})$ is true
  - $\forall x \forall y (x \wedge y) \vee (\bar{x} \wedge \bar{y})$ is false

- $SAT \leftrightarrow \exists x_1 \dots \exists x_n \varphi(x_1, \dots, x_n)$ is true

PS-completeness: example

- Quantified Boolean formula (QBF), prenex form:
- $z \in \{0,1\}^n;\ Q_i \in \{\forall, \exists\}; \varphi(z) = Q_1 z_1 \dots Q_n z_n f(z)$
- Ie: $\forall x \exists y (x \wedge y) \vee (\bar{x} \wedge \bar{y})$ is true, $\forall x \forall y (x \wedge y) \vee (\bar{x} \wedge \bar{y})$ is false
- TQBF – {all true QBF}
- TQBF – is PS-complete
    - N variables, M-sized $f(z)$, including constants
    - $s(n, m) = s(n-1, m) + O(m)$
    - $\forall L \in PS, L \leq_p TQBF$
- Savitch's theorem: $\forall^* S: \mathbb{N} \to \mathbb{N}, S(n) \geq \log n \ \to NPS(S(n)) \subseteq PS(S(n)^2)$

Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities.
https://www.sciencedirect.com/science/article/pii/S002200007080006X/pdf?md5=8848287d0e96ddfb587edd4eda925720&pid=1-s2.0-S002200007080006X-main.pdf

PS-completeness and winning strategy

- 2 players, perfect information games
- $\exists z_1 \forall z_2 \exists z_3 \forall z_4 \ldots f(z)$

NL-completeness

- Reduction:
- A, B – decision problems. A is log space reducible to B ($A \leq_{log} B$) if $\exists f$ computable in log space, $x \in A \leftrightarrow f(x) \in B$ & $B \in \boldsymbol{L}$

NL-completeness

- Reduction:
- A, B – decision problems. A is log space reducible to B ($A \leq_{log} B$) if $\exists f$ computable in log space, $x \in A \leftrightarrow f(x) \in B$ & $B \in \boldsymbol{L}$

- $B \in \boldsymbol{L}$ & $A \leq_{log} B \rightarrow A \in \boldsymbol{L}$
- Composition: M_f and M_B

NL-completeness

- Reduction:
- A, B – decision problems. A is log space reducible to B ($A \leq_{log} B$) if $\exists f$ computable in log space, $x \in A \leftrightarrow f(x) \in B$ & $B \in \boldsymbol{L}$
- A is **NL**-hard if $\forall B \in \boldsymbol{NL} \ \rightarrow B \leq_{log} A$
- A is **NL**-complete if $A \in \boldsymbol{NL}$ and A is **NL**-hard

NL-completeness

- Reduction:
- A, B – decision problems. A is log space reducible to B ($A \leq_{log} B$) if $\exists f$ computable in log space, $x \in A \leftrightarrow f(x) \in B$ & $B \in \boldsymbol{L}$
- A is **NL**-hard if $\forall B \in \boldsymbol{NL} \rightarrow B \leq_{log} A$
- A is **NL**-complete if $A \in \boldsymbol{NL}$ and A is **NL**-hard
- $\boldsymbol{NL} \subseteq \boldsymbol{P}$
  - $2^{O\left(S(n)\right)} = n^{O(1)}$ configurations {C}
  - $G = \langle C, \delta \rangle, |C| = O(n^c)$

NL-completeness

- $STCONN = \{\langle G, s, t\rangle, \exists path\ s \to t\}$ is NL-complete
- $STCONN \in NL$
- $STCONN$ is NL-hard

Summary

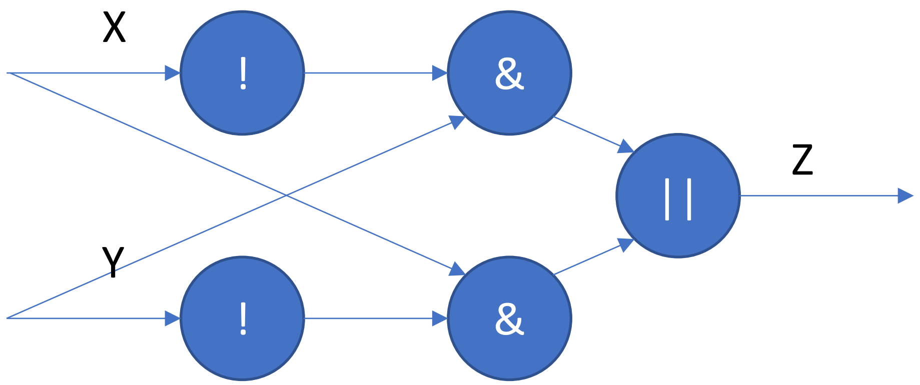
- $L \subseteq NL \subseteq P \subseteq NP \subseteq PS \subseteq EXP \subseteq NEXP$

Previous results

- PSPACE, L, NL classes and their completeness
- QBF, PS-complete TQBF problem
- NL-complete connectivity problem
- $L \subseteq NL \subseteq P \subseteq NP \subseteq PS \subseteq EXP \subseteq NEXP$

Boolean circuits

- Directed acyclic graph with n srcs and 1 sink



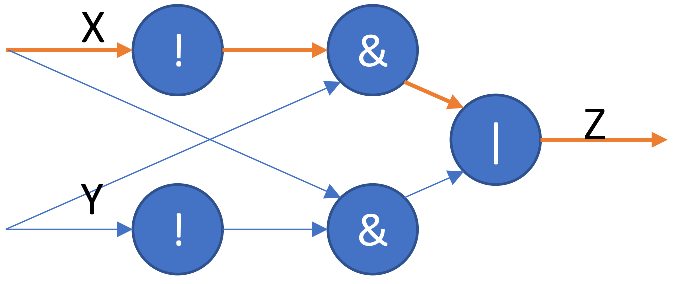
- $\{C_n\}, n \in \mathbb{N}$: $\forall x \in \{0,1\}^n, x \in L \leftrightarrow C_n(x) = 1$

```
bool xor2(bool X, bool Y)
{
	bool nx = !X;
	bool ny = !Y;
	bool z1 = nx && Y;
	bool z2 = ny && X;
	return z1 || z2;
}
```

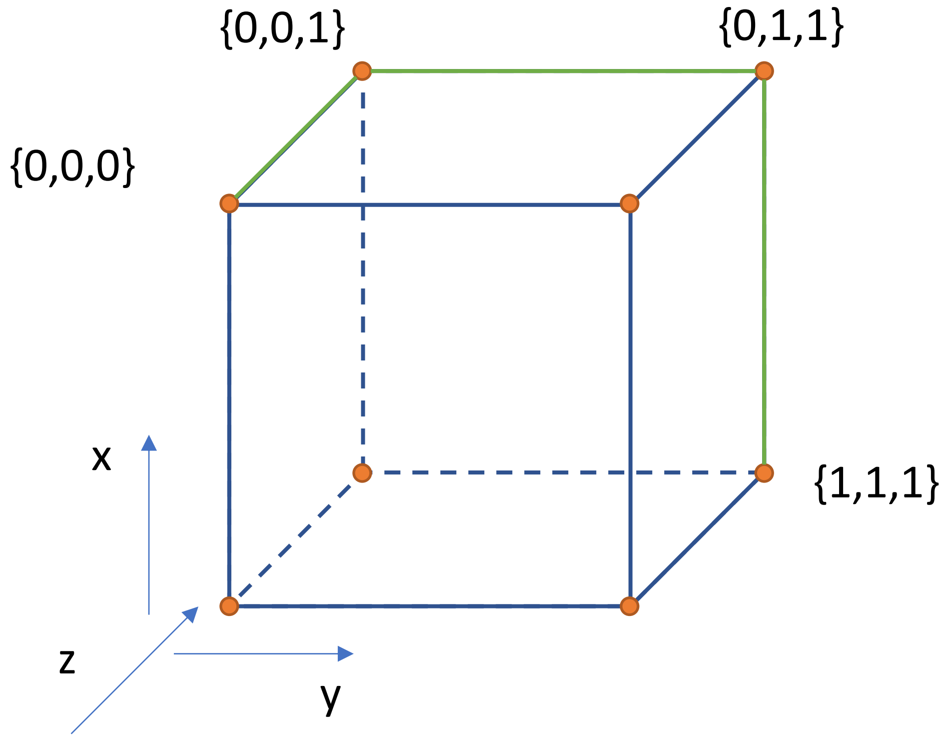
Boolean circuits

- $CSAT = \{circuits\ reps{:}\ \exists s = \{0,1\}^n\ s.t.\ C(s) = 1\}$ - NP-complete
- $f{:}\,\{0,1\}^n \to \{0,1\}$ solvable in $O(2^n/n)$ size.
- $P_{/poly}$ - languages, decidable by polynomial-sized circuits
- $P \subseteq P_{/poly}$
- Karp-Lipton theorem: $NP \subseteq P_{/poly}$ is unlikely
- $L \in NC \leftrightarrow \exists efficient\ parallel\ alg$

Depth



Massively parallel computers



- $O(\log n)$ steps communication
- Small amount of work per node per step: $O(\log n)$ bits
- Efficient: $n^{O(1)}$ nodes & $T(n) = (\log n)^{O(1)}$
- Example: carry lookahead adder

Probabilistic Turing machines

- Probabilistic TM (PTM): δ_0, δ_1 chosen with ½ prob.
- $T: \mathbb{N} \to \mathbb{N}, L \subseteq \{0,1\}^*$, PTM decides $L\ in\ T(n)$: $\forall x \Pr[M(x) = L(x)] \geq 2/3$
- $BPTIME\big(T(n)\big)$ - decided in T(n)
- $\boldsymbol{BPP} = \bigcup_c BPTIME(n^c)$
- $BPP \subseteq EXP$
- $BPP = P?$
- $P \subseteq BPP \subseteq P_{/poly}$

Error reduction: PTM robustness

- $BPP_{1/2+n^{-c}}$: $L \subseteq \{0,1\}^*$, PTM M decides $L\ in\ T(n)$: $\forall x$: $\Pr[M(x) = L(x)] \geq 1/2 + |x|^{-c}, c > 0$.
- For any x $\in \{0,1\}^*$ $\exists PTM\ M'$: $\Pr[M'(x) = L(x)] \geq 1 - 2^{-|x|^d}$
- M' runs $M(x)$ for $8|x|^{2c+d}$ times, output decided on majority of 1's and outputs $y_1 \dots y_k \in \{0,1\}$
- Random independent vars $X_i = 1\ iff\ y_i = L(x)$: $E[X_i] = \Pr[X_i = 1] \geq p, p = \frac{1}{2} + |x|^{-c}$; Chernoff bound gives:
- $\Pr\left[\left|\sum_{i=1}^{k} X_i - pk\right| > \delta pk\right] < e^{-\frac{\delta^2}{4}pk}$, "$\delta \to 0$".
- Set $\delta = \frac{|x|^{-c}}{2} \to e^{-\frac{1}{4|x|^{2c}}*\frac{1}{2}8|x|^{2c+d}} \leq 2^{-|x|^d}$

Median example

- $\{x_0, \dots, x_{n-1}\}$
- Find k-th element(k, $x_0, \dots\ x_{n-1}$):

1. Choose $i \in \{n\} \to b = x_i$
2. Go through the set & count $m = |\{x_i \leq b\}|$
3. If m=k – done
4. m>k, Find k-th element(k, $\{x_i \leq b\}$)
5. Else, Find k-th element(k-m, $\{x_i > b\}$)

Median example

- $\{x_0, \ldots, x_{n-1}\}$
- Find k-th element(k, $x_0, \ldots\ x_{n-1}$):

1. Choose $i \in \{n\} \rightarrow b = x_i$
2. Go through the set & count $m = |\{x_i \leq b\}|$
3. If m=k – done
4. m>k, Find k-th element(k, $\{x_i \leq b\}$)
5. Else, Find k-th element(k-m, $\{x_i > b\}$)

$$T(n) = cn + \frac{cn}{2} + \frac{cn}{4} + \frac{cn}{8} + \cdots = 2cn \sim O(n)$$

*Induction: assume $T(n) \leq 10cn$

Deterministic O(n) algorithm: http://people.csail.mit.edu/rivest/pubs/BFPRT73.pdf

Quantum computation

- Qubit: $\alpha_0|0\rangle + \alpha_1|1\rangle$
- 2 qubit system state: $\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$
- $v = \langle v_{0^n}, v_{0^{n-1}1}, \dots, v_{1^n}\rangle$, $F(v) = \sum_x v_x F(|x\rangle)$
- Swap: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $(|01\rangle \to |10\rangle)$
- Copy? CNOT: $|xy\rangle = |x(x \oplus y)\rangle$

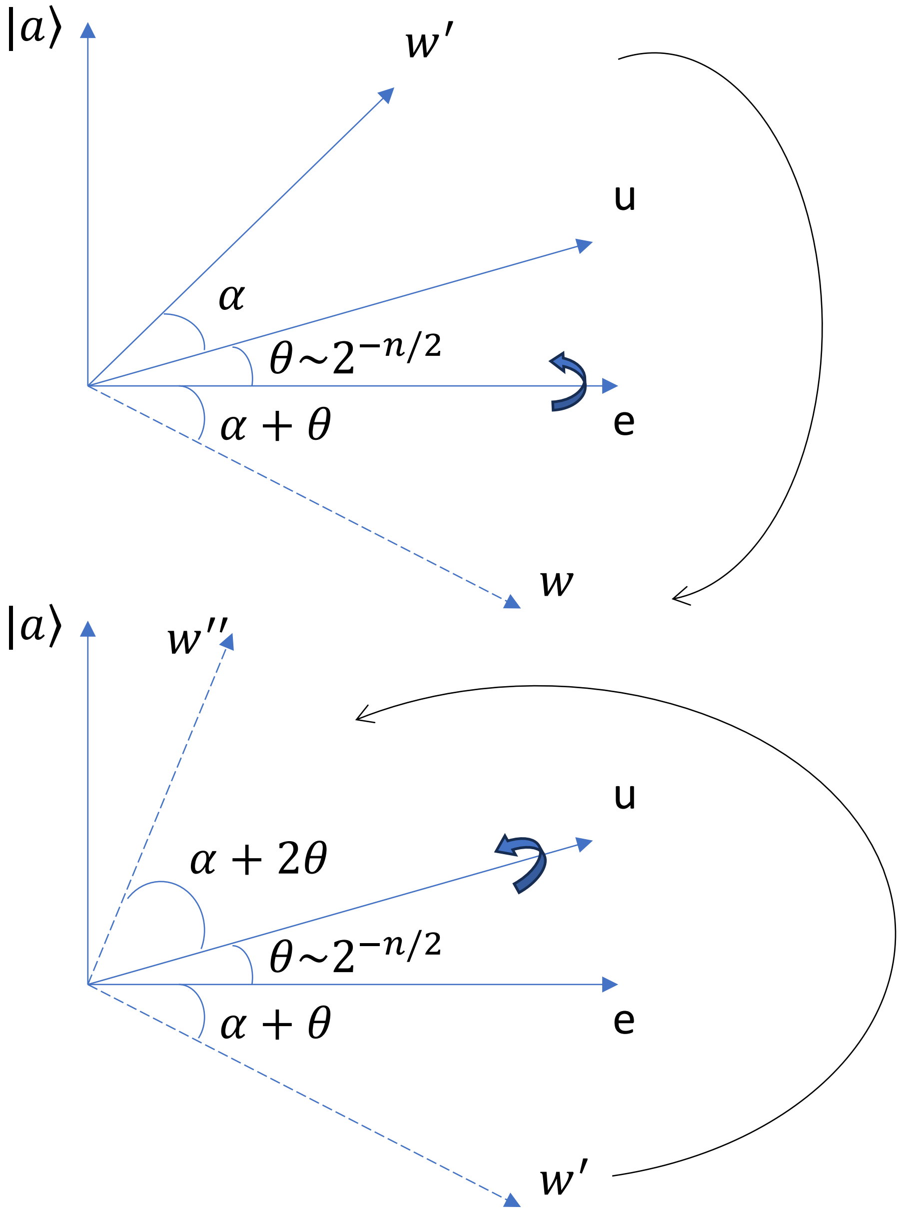
https://www.nature.com/articles/s41586-021-03928-y

Quantum computation: BQP

- f is computable in quantum $T(n)$ if there's TM$(1^n, 1^{T(n)})$ $\forall n$ outputs n gates descriptions F_i that compute $f(x)$ with 2/3 prob.
  - Init with $|x0^{n-m}\rangle$
  - Apply Fs
  - Measure and output reg value
- $f: \{0,1\}^* \to \{0,1\}, f \in \boldsymbol{BPQ}\ if\ \exists p - polynomial$, so that f is computable in quantum $p(n)$ time

Grover's search algorithm

- There is a quantum algorithm that for poly-time computable function $f: \{0,1\}^* \rightarrow \{0,1\}$ finds a such that $f(a) = 1$ in $poly(n)2^{n/2}$ time.
- Rotate a state vector (of a register) to unknown $|a\rangle$ taking two reflections around uniform state vector an orthogonal e $= \sum_{x \neq a} |a\rangle$ (Hadamard)
- Each rotation goes from $\frac{\pi}{2} - \alpha$ to $\frac{\pi}{2} - \alpha - 2\theta$
- In $O\left(\frac{1}{\theta}\right) = O(2^{\frac{n}{2}})$ steps the resulting vector inner product with $|a\rangle$ yields a with probability ¼



*requires n+1+m register size (m is scratch)

Integer factorization: Shor’s algorithm

- Find the smallest r such that $A^r \equiv 1 \ (mod\ N)$ for a random A
- The order r will be even and $A^{r/2} - 1$ will have a nontrivial common factor with N with high prob.
- Fast exponentiation can be polynomial with a classical machine
- The algorithm translates initial zero state into $|x\rangle, x \leq N, A^x \equiv y \ (mod\ N)$ for some random $y \leq N - 1$.
- Sequence of these states produce an arithmetic progression $x_0 + ri, i = 1,2, \ldots,$ where $A^{x_0} \equiv y \ (mod\ N)$
- Obtaining the period can be done via Quantum Fourier Transform (QFT) in log^2

MIT 18.435

Quantum computation: BQP

- $P \subseteq BPQ$ – Boolean circuits are a subcase of quantum circuits.
- $BPP \subseteq BPQ$ – using a universal basis for quantum operations.
- $BPQ \subseteq PSPACE$

PCP theorem

- Proof system
  - Boolean formula
- Verify a certificate by checking random constant number of locations
  - Correct certificate never fails to convince
  - Guarantees to reject with high prob. for any unsatisfiable formula

- Approximations are not easier than exact solutions
- ρ-approximation
- MAX-3SAT

$$\mathbf{NP} = \mathbf{PCP}[O(\log n), O(1)]$$

Resources

- Computational Complexity: A Modern Approach (https://theory.cs.princeton.edu/complexity/book.pdf)
- Introduction to Algorithms, Cormen (i.e. https://web.ist.utl.pt/~fabio.ferreira/material/asa/clrs.pdf)
- Classical Mathematical Logic: The Semantic Foundations of Logic (more on Boolean formula normal forms)

Backup