A Distributed PCP Theorem for Hardness of Approx. in P

Amir Abboud (IBM Almaden)

Aviad Rubinstein (Harvard → Stanford)

Ryan Williams (MIT)

Hardness of Approx. in P

Example application: **Max Inner Product**

Input: $A, B \subset \{0,1\}^d$

Output: $\max_{\substack{a \in A \\ b \in B}} a \cdot b$

- Also known as: Max Intersection, Jaccard Similarity Search, Tanimoto Similarity Search, (Batch) Subset Queries, Partial Matching, …
- Other results: LCS Closest Pair, Approximate Regular Expression Matching, Diameter in product metric, …

* an apology…

	Bichromatic	vs	Monochromatic
Input:	$A, B \subset \{0,1\}^d$		$U \subset \{0,1\}^d$
Output:	$\max_{\substack{a\in A \\ b\in B}} a \cdot b$		$\max_{\substack{u,v\in U \\ u\neq v}} u \cdot v$

- Bichromatic more important (implies hardness of data structures)
- In FOCS'17 proceedings, we claimed our techniques also apply to the monochromatic case. We were wrong. **Sorry**!!
- **Monochromatic case: still open** (even for exact!)
  See e.g. [David, C.S., Laekhanukit '18], [Williams '18]
- Open problem: why do we still have published proceedings?

Hardness of Approx. in P

Example application: **Max Inner Product**

Input: $A, B \subset \{0,1\}^d$

Output: $\max_{\substack{a \in A \\ b \in B}} a \cdot b$

- Also known as: Max Intersection, Jaccard Similarity Search, Tanimoto Similarity Search, (Batch) Subset Queries, Partial Matching, …
- Other results: LCS Closest Pair, Approximate Regular Expression Matching, Diameter in product metric, …

Hardness of Approx. in P

- Brute force algorithm: $O(N^2 d)$ time.
- **Can we do truly faster: $O(N^{2-\varepsilon})$?**

- Assuming Strong Exponential Time Hypothesis (SETH), no $O(N^{2-\varepsilon})$-time algorithms for:

SETH & PCP:
A 4-minute crash course

(Buckle your seatbelts…)

SETH-hardness reductions [Williams'05]

- Strong Exponential Time Hypothesis (SETH):

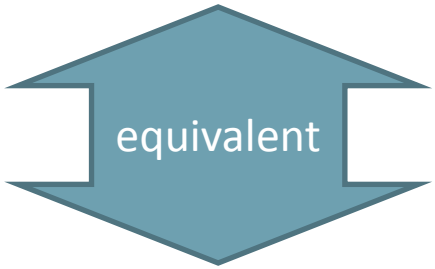
  k-SAT requires time $2^n \approx N^2$

- The reduction:
  - Input: φ on n variables
  - Construct a vector $a \in A$ for each $\alpha = (x_1 \dots x_{\frac{n}{2}}) \in \{0,1\}^{n/2}$
  - Construct a vector $b \in B$ for each $\beta = (x_{\frac{n}{2}+1} \dots x_n) \in \{0,1\}^{n/2}$
  - Gadgets: $a \cdot b$ large $\leftrightarrow$ $x = (\alpha;\beta)$ satisfies φ

- Reduction size: $N \approx 2^{n/2}$

The PCP Theorem [AS'98, ALMSS'98]

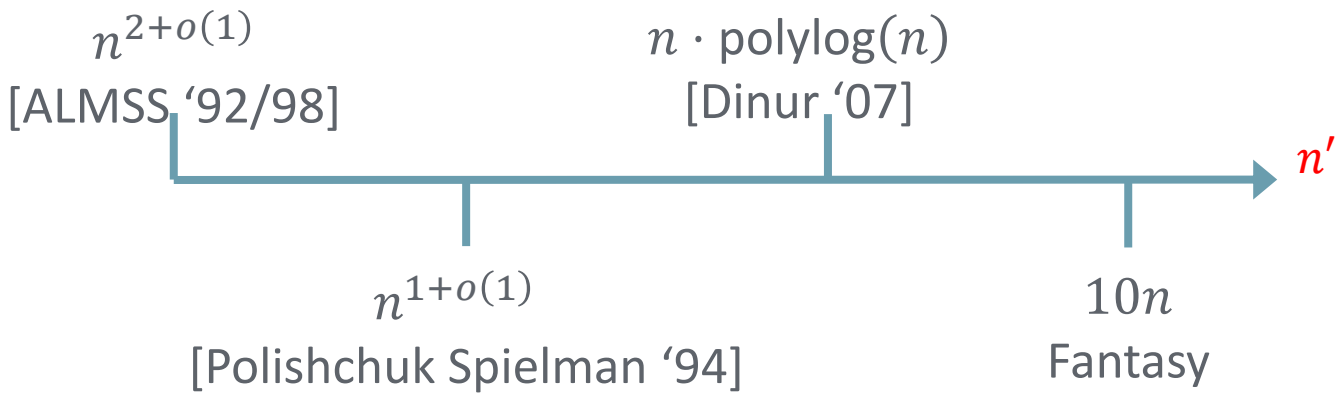
- Probabilistically Checkable Proofs:
  - 3-SAT φ + sat. assignment $x \in \{0,1\}^n \rightarrow$ proof $\pi(x)$
  - Verifier only reads a small number of bits from $\pi(x)$!



- Hardness amplification
  - 3-SAT φ on n variables → 3-SAT φ' on n' variables
  - Approximating φ' is as hard as solving φ exactly!

The PCP Blowup

3-SAT φ on n variables → 3-SAT φ' on n' variables



Hardness in P from PCP+SETH?

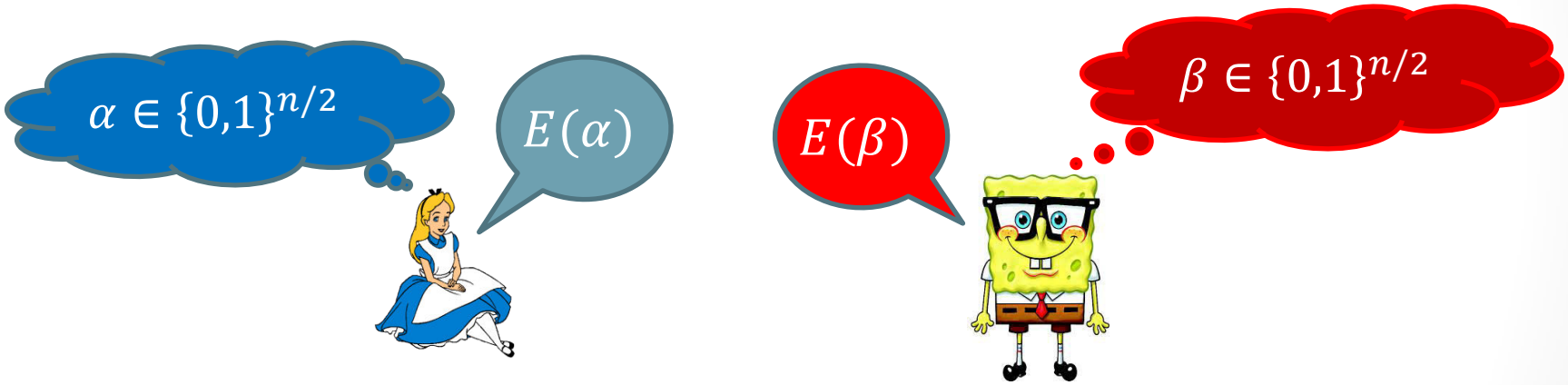
- Reduction size $N \approx 2^{n'/2} = 2^{5n}$ (with fantasy PCP)
- Then, running time of $N^{1.1} \approx 2^{5.5n}$ refutes NOTHING

A new framework

(More applications to follow…)

Distributed PCP (finally!)

Warmup: distributed error correcting encoding



Now $\big(E(\alpha); E(\beta)\big)$ is a pretty good encoding of $x = (\alpha; \beta)$ ☺

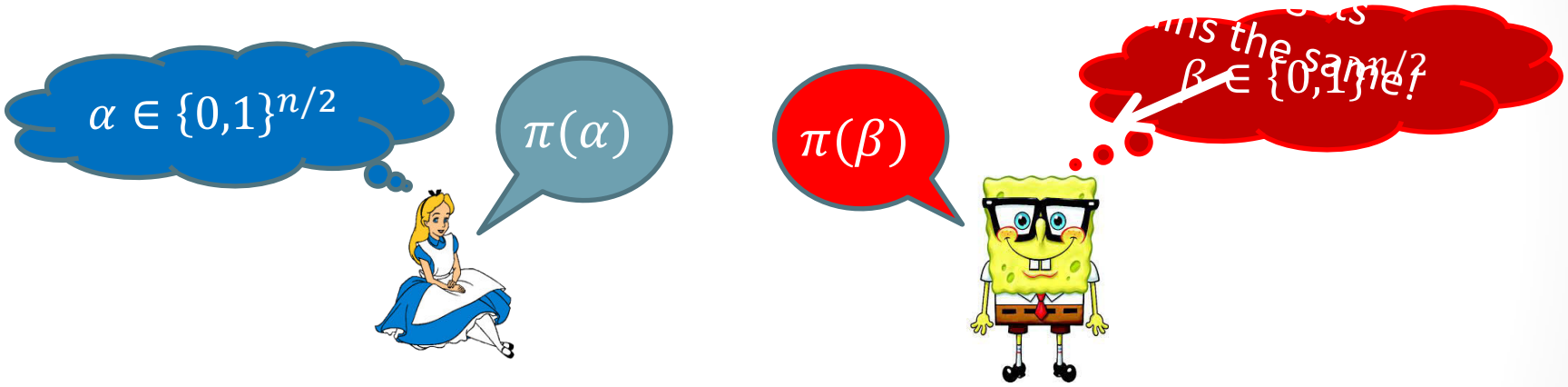
Can we do the same for PCP?

Reminder: PCP Theorem

- Probabilistically Checkable Proofs:
  - 3-SAT φ + sat. assignment $x \in \{0,1\}^n \rightarrow$ proof $\pi(x)$
  - Verifier only reads a small number of bits from $\pi(x)$!

Distributed PCP (finally!)

Everyone knows the 3-SAT φ (over n variables)



Now $\big(\pi(\alpha);\pi(\beta)\big)$ is a pretty good PCP of
"$(\alpha;\beta)$ satisfies φ"

Distributed PCP (finally!)

Every

The reduction:

Input: φ on n variables

Construct a vector $a \in A$ for each $\boldsymbol{\pi(\alpha)}$

Construct a vector $b \in B$ for each $\boldsymbol{\pi(\beta)}$

Gadgets: $a \cdot b \approx \Pr[\text{verifier accepts } (\pi(\alpha); \pi(\beta))]$

# of gadgets remains the same!

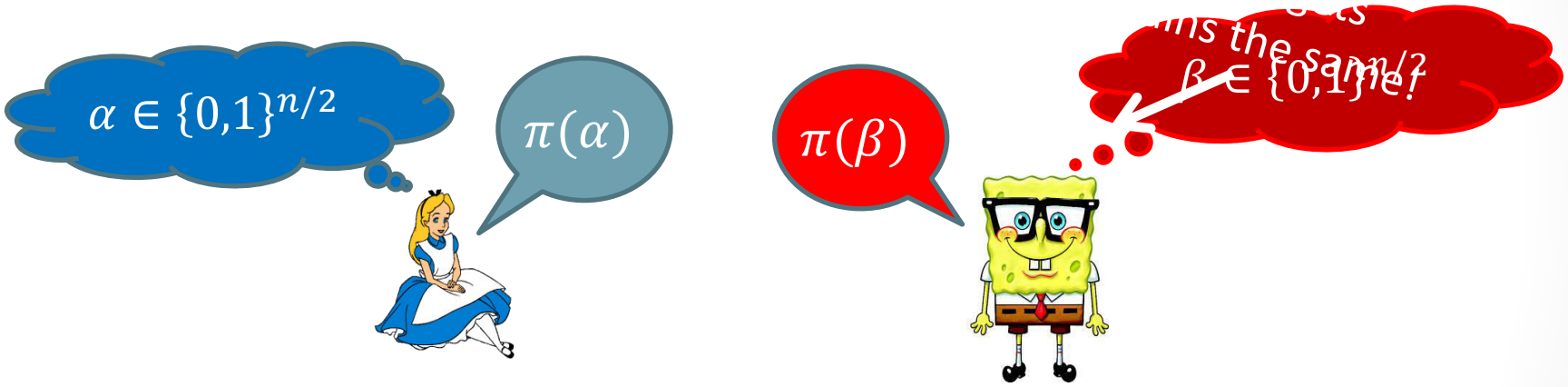
$(\alpha; \beta$

Good news: completely circumvents the PCP blowup ☺

But… can we actually construct distributed PCPs?

Distributed PCP (finally!)

Everyone knows the 3-SAT φ (over n variables)



Now $\big(\pi(\alpha);\pi(\beta)\big)$ is a pretty good PCP of
"$(\alpha;\beta)$ satisfies φ"

Good news: completely circumvents the PCP blowup ☺

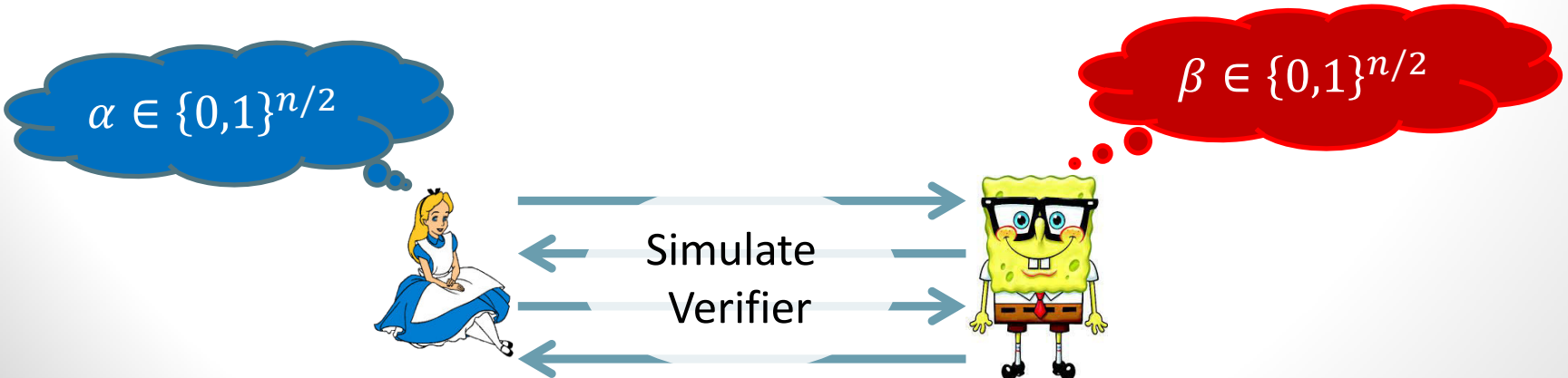
But… can we actually construct distributed PCPs?

Distributed PCP don't exist (what?!)

Communication Complexity:
Reduction from **Set Disjointness** [Reingold'17]

$$\varphi = \bigwedge (\neg \alpha_i \vee \neg \beta_i)$$

(α, β) satisfy $\varphi \Leftrightarrow \alpha, \beta$ disjoint.



Merlin-Arthur-Bob Communication

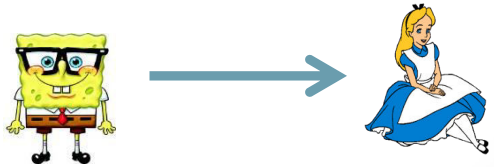
Merlin: Omniscient, untrusted prover/witness

MA protocol for Set-Disjointness [Aaronson-Wigderson '09]:

1. Merlin sends Alice $\widetilde{\boldsymbol{O}}(\sqrt{\boldsymbol{n}})$ bits (μ)

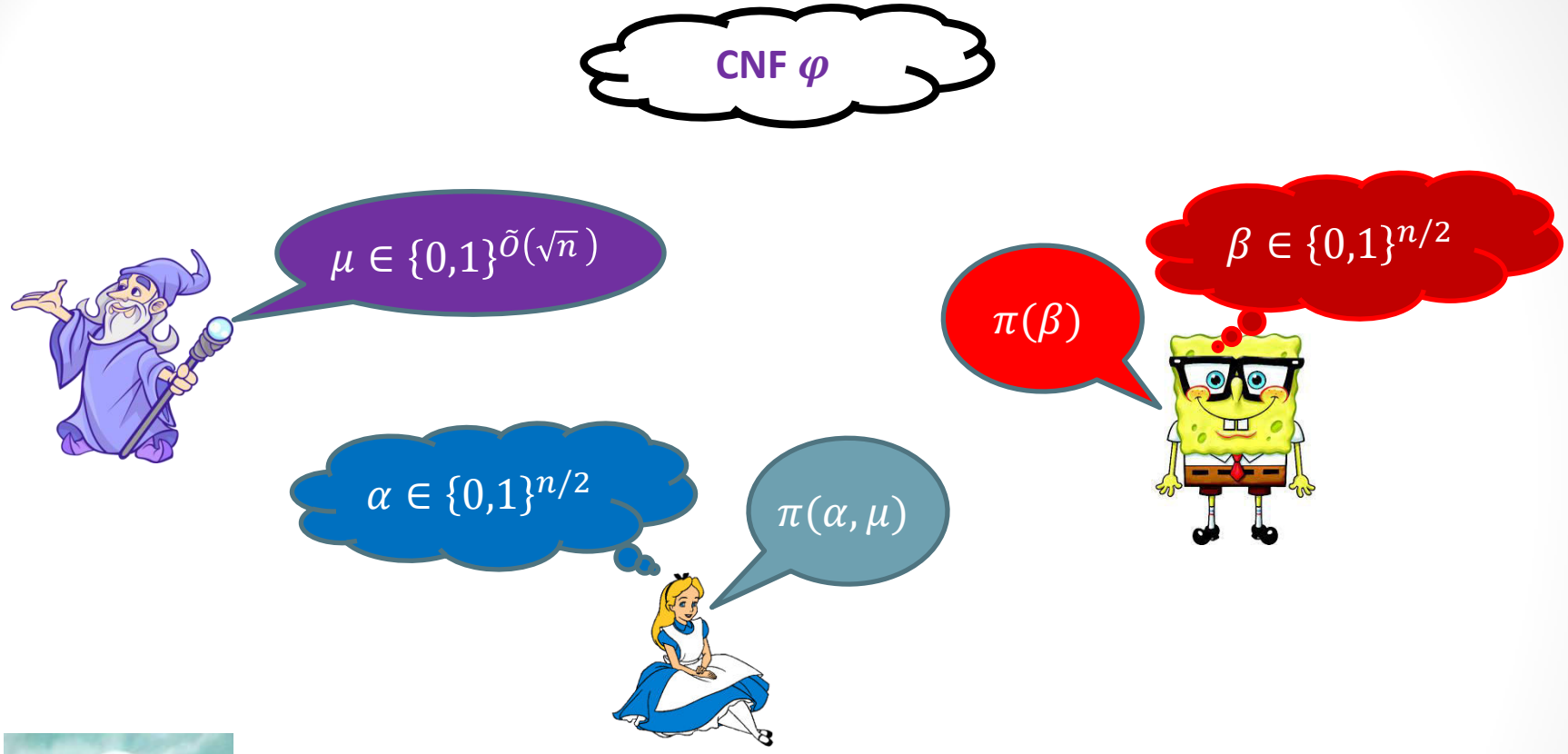
2. Alice & Bob toss $\log n$ coins

3. Bob sends Alice $\widetilde{\boldsymbol{O}}(\sqrt{\boldsymbol{n}})$ bits



4. Alice Accepts/Rejects

Non-Deterministic & Distributed PCP



The Verifier:

- Reads one symbol ($\tilde{O}(\sqrt{n})$ bits) from $\pi(\alpha, \mu), \pi(\beta)$
- Outputs accept/reject

The full reduction

MA Set Disjointness Protocol

1. **Merlin sends Alice μ**
2. **A & B toss coins r**
3. **Bob sends Alice k^***
4. **Alice accepts/rejects**

Input: CNF φ

Output: $A, B \subset \{0,1\}^d$, for $d = 2^{\log n} \times 2^{\sqrt{n}}$

- For every $\beta \in \{0,1\}^{n/2}$, construct a vector $b^\beta \in B$:
  - For every coin toss $r \in \{0,1\}^{\log n}$
  - For every message $k \in \{0,1\}^{\sqrt{n}}$
  - $\left[b^\beta\right]_{r,k} = 1 \Leftrightarrow$ Bob(β, r) outputs message k

- For every $\alpha \in \{0,1\}^{n/2}$ and $\mu \in \{0,1\}^{\sqrt{n}}$, construct a vector $a^\alpha \in A$:
  - $[a^{\alpha,\mu}]_{r,k} = 1 \Leftrightarrow$ Alice(α, μ, r) accepts message k from Bob

Then: $a^{\alpha,\mu} \cdot b^\beta = \Pr[\text{Alice accepts in MA protocol}]$

QED

My life will never be the same

Applications, extensions, and lots of *open problems*!

What's your favorite CC model?

CC model	Computational application	Reference
Merlin Arthur (MA)	Max-IP, ANN, …	[ARW'17] [CLM'18] [R'18] [Chen'18]

What's your favorite CC model?

CC model	Computational application	Reference
Merlin Arthur (MA)	Max-IP, ANN, …	[ARW'17] [CLM'18] [R'18] [Chen'18]
NP·UPP	Max-IP, … (small dimension)	[Williams'18] [Chen'18]
Quantum	$\{\pm 1\}$-Max-IP	[Chen'18]

What's your favorite CC model?

CC model	CC problem	Computational application	Reference
Merlin Arthur (MA)	Set Disjointness	Max-IP, ANN, …	[ARW'17] [CLM'18] [R'18] [Chen'18]
NP·UPP	"	Max-IP, … (small dimension)	[Williams'18] [Chen'18]
Quantum	"	$\{\pm 1\}$-Max-IP	[Chen'18]

What's your favorite CC model?

CC model	CC problem	Computational application	Reference
Merlin Arthur (MA)	Set Disjointness	Max-IP, ANN, …	[ARW'17] [CLM'18] [R'18] [Chen'18]
NP·UPP	"	Max-IP, … (small dimension)	[Williams'18] [Chen'18]
Quantum	"	$\{\pm 1\}$-Max-IP	[Chen'18]
Randomized (RP)	Equality	k-Dominating Set (Parametrized)	[CLM'18]

What's your favorite CC model?

CC model	CC problem	Computational application	Reference
Merlin Arthur (MA)	Set Disjointness	Max-IP, ANN, …	[ARW'17] [CLM'18] [R'18] [Chen'18]
NP·UPP	"	Max-IP, … (small dimension)	[Williams'18] [Chen'18]
Quantum	"	$\{\pm 1\}$-Max-IP	[Chen'18]
Randomized (RP)	Equality	k-Dominating Set (Parametrized)	[CLM'18]
Interactive (IP)	Circuit-SAT	LCS, …	[AR'18] [CGLRR]

Amazing Questions

- Longest Common Subsequence with two (long) strings
  - # of clauses vs # of assignments
  - No *deterministic* approximation algorithms [AB'17][A**R**'18][CGLR**R**'18]
- Approximate Closest Pair: Edit Distance vs ℓ_2
  - For edit distance the known hardness [**R**'18] is far from algorithms!
- $(1+\varepsilon)$-Approximate Dynamic Maximum Matching
  - For exact, known reductions from 3-SUM, Triangle Detection, and OMV
- A quantum analog for SETH?
- New PCP-like models