Strong ETH Breaks With Merlin and Arthur

Or: Short Non-Interactive Proofs of Batch Evaluation

Ryan Williams Stanford

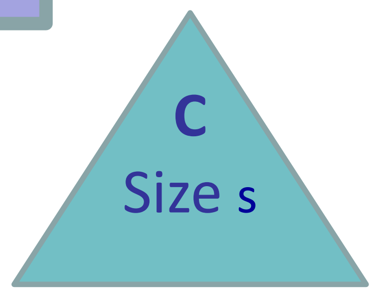
Two Stories

<u>Story #1:</u> The Circuit and the Adversarial Cloud.

Let me do it!

Given: $a_1, \ldots, a_K \in F^n$
Want: $C(a_1), \ldots, C(a_K) \in F$

Would naively take
$\sim s \cdot K$ time



n variables

C = **Arithmetic Circuit** over +, × in field F

Computes some polynomial

$C(x_1, \ldots, x_n) \in F[x_1, \ldots, x_n]$

of degree d

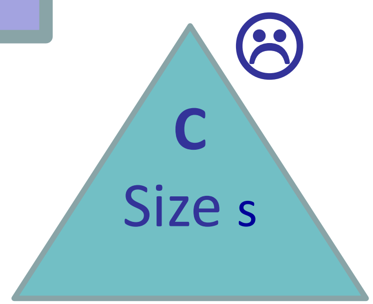
Two Stories

Story #1: The Circuit and the Adversarial Cloud.

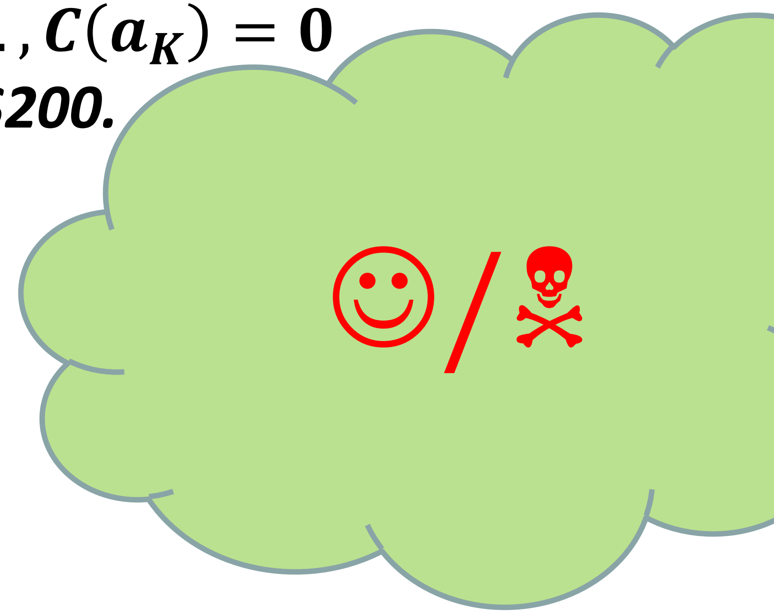
$\boldsymbol{C(a_1) = 0, \dots, C(a_K) = 0}$

You owe me $200.

Given: $a_1, \dots, a_K \in F^n$
Want: $C(a_1), \dots, C(a_K) \in F$



n variables



C = **Arithmetic Circuit** over +, × in field F

Computes some polynomial

$C(x_1, \dots, x_n) \in F[x_1, \dots, x_n]$

of degree d

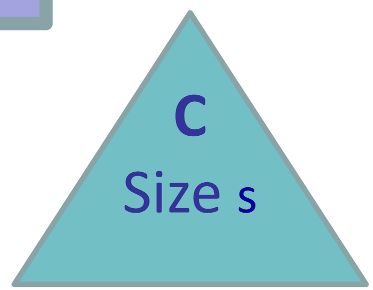
Two Stories

Story #1: The Circuit and the Adversarial Cloud.

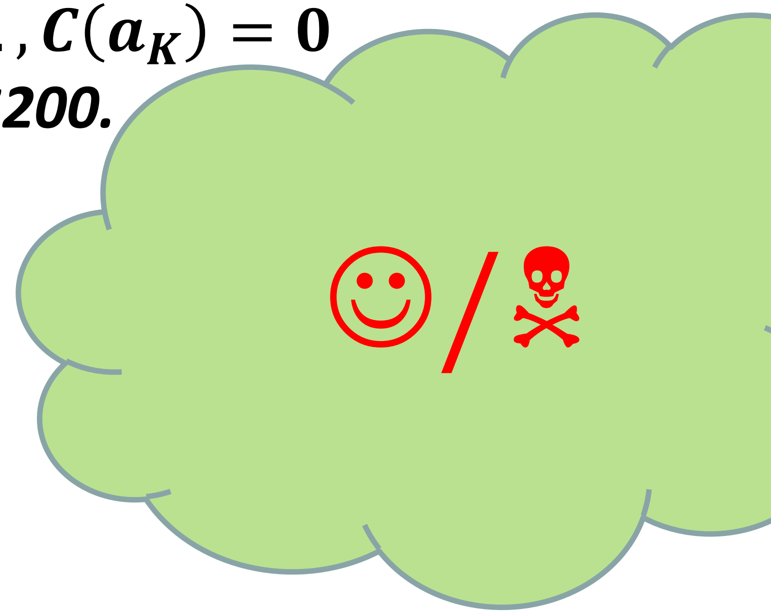
$\boldsymbol{C(a_1) = 0, \ldots, C(a_K) = 0}$

You owe me $200.

Given: $a_1, \ldots, a_K \in F^n$
Want: $C(a_1), \ldots, C(a_K) \in F$



n variables



How can C *very quickly* check the work of the Cloud?
(without just doing the work himself…)

IP = PSPACE [LFKN’92, Shamir’92] - intractable provers

Delegating Computation [GKR’08,…,RRR’16]

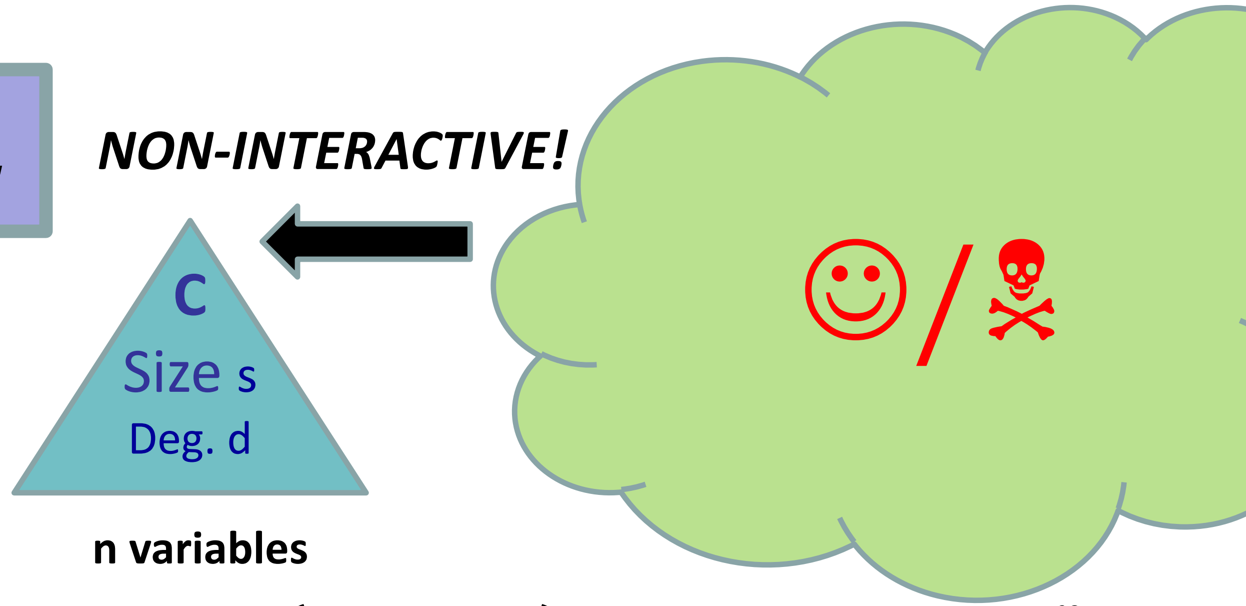
O(1)-round interactive proofs for low-depth ckts with fast verification

Two Stories

<u>Story #1:</u> The Circuit and the Adversarial Cloud.

Given: $a_1, \dots, a_K \in F^n$
Want: $C(a_1), \dots, C(a_K) \in F$

Note the input itself is $\approx (\boldsymbol{K \cdot n + s})$ size…



Thm: $\exists$ **Verifier** alg. where for all $C(x_1, \dots, x_n)$ and $a_1, \dots, a_K \in F^n$,

- There is a ***proof*** of length $\boldsymbol{\tilde{O}(K \cdot d)}$ the **Cloud** can send, along with values $C(a_1), \dots, C(a_K) \in F$, such that…
- The **Verifier** can toss $\boldsymbol{poly\left(\log\left(\frac{dK|F|}{\epsilon}\right)\right)}$ coins and ***check the proof*** in about $\boldsymbol{\tilde{O}(K \cdot (n + d) + s)}$ time, with probability of error $\boldsymbol{\leq \epsilon}$.

Two Stories

Story #2: The Many Exponential Time Hypotheses.

Once upon a time…


3-SAT = { satisfiability of formulas in CNF, all clauses have at most 3 literals}

Exponential Time Hypothesis (ETH) [IPZ'01] : 3-SAT requires at least $2^{\alpha n}$ (randomized) time, for some $\alpha > 0$

(n = # of variables in the formula)

Best known 3-SAT algorithm: about 1.31^n time

Vast strengthening of $P \neq NP$

Many Exponential-Time Lower Bounds, assuming ETH!

Assuming ETH, the problems Independent Set, Clique, Vertex Cover, Dominating Set, Graph Coloring, Max Cut, Set Splitting, Hitting Set, Min Bisection, Feedback Vertex Set, Hamiltonian Path, Max Leaf Spanning Tree, Subset Sum, Knapsack, 3-Dimensional Matching, Cluster Editing, Treewidth *and many others* do not have $2^{o(n)}$ time algorithms.

(Note: Not easy! These do not follow from typical NP-completeness reductions)

Could We Predict the *Exact* Exponents of Running Times?

Find evidence that 3-SAT is not in 1.0000001^n time?

(Note for $n \leq 10^8$, 1.0000001^n is pretty small!)

Assuming ETH, we can distinguish between runtimes like 2^n and $2^{n/\log n}$, but not between 1.2^n and 1.3^n

Need a stronger ETH ...

Best known k-SAT algorithms: **$2^{n - n/O(k)}$ time…**

Achieved by four different algorithmic paradigms!

Strong Exponential Time Hypothesis (SETH) [IP'01,CIP'09]

SETH: For every $\varepsilon > 0$ there exists a $k \geq 3$ such that ***Satisfiability of k-CNFs*** requires $(2-\varepsilon)^n$ time.

Theorem: SETH $\Rightarrow$ ETH

An even more productive hypothesis!

Tight lower bounds on **many** natural polynomial-time tasks, edit distance **[BI'15]**, LCS **[ABV'15]**, dynamic graph algorithms **[AV'14]**, … many refs very recently!

Refuting **SETH → new circuit lower bounds [JMV]**

(Valiant series-parallel circuits)

Nondeterministic Strong ETH (NSETH) [CGIMPS'16]

NSETH: For every $\varepsilon > 0$ there exists a $k \geq 3$ such that ***refuting unsatisfiable k-CNFs*** requires $(2-\varepsilon)^n$ ***nondeterministic*** time.

We cannot (yet?) refute this.

But we can lay some other conjectures to rest…

Non-Interactive Proofs of UNSAT That Beat Exhaustive Search

Thm: There is a proof system for UNSAT such that ***every UNSAT Boolean formula of poly(n) size*** has a proof of length $\sim 2^{n/2}$ verifiable with ***poly(n) bits of randomness*** in $\sim 2^{n/2}$ time.

(Can even count SAT assignments)

If this proof system can be "de-randomized" then NSETH is false!

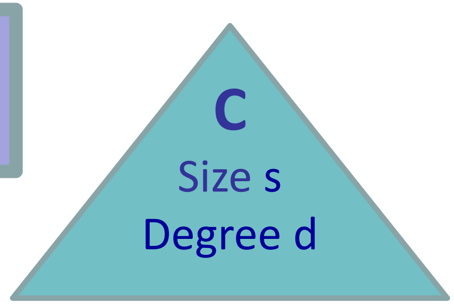
The derandomization problem reduces to an interesting *univariate* polynomial identity test

Thm: For all $C(x_1, \ldots, x_n)$ and all $a_1, \ldots, a_K \in F^n$, there is a **Verifier** alg:

- There is a proof of length $\tilde{O}(K \cdot d)$ the **Cloud** can send, along with values $C(a_1), \ldots, C(a_K) \in F$, such that
- the **Verifier** can toss $poly\left(\log\left(dK\frac{|F|}{\epsilon}\right)\right)$ coins and ***check the proof*** in about $\tilde{O}(K \cdot (n + d) + s)$ time, with probability of error $\leq \epsilon$.

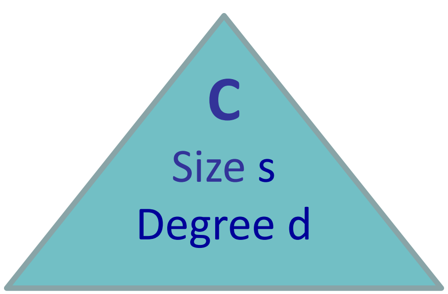
Given: $a_1, \ldots, a_K \in F^n$
Want: $C(a_1), \ldots, C(a_K) \in F$

Naively takes
$\sim s \cdot K$ time… get $\sim s + K$



n variables

Idea: Define a *univariate* polynomial that simulates evaluation of C on the a_i



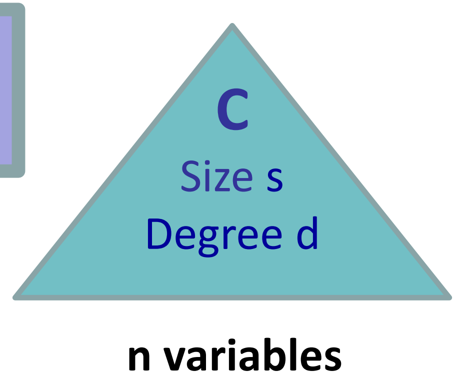
n variables

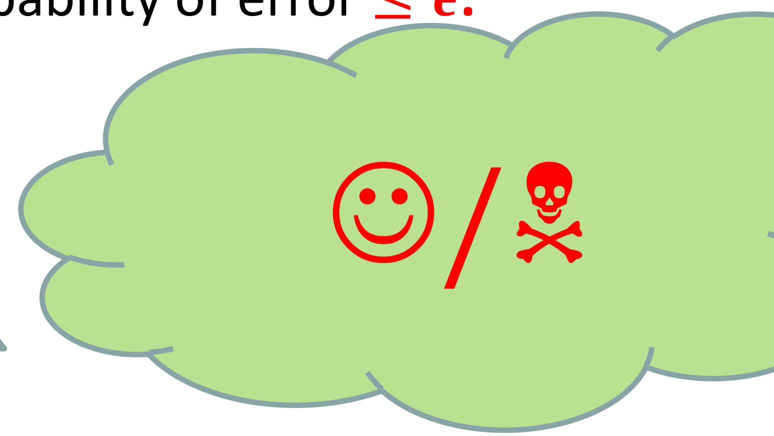
Thm: For all $C(x_1, \dots, x_n)$ and all $a_1, \dots, a_K \in F^n$, there is a **Verifier** alg:

- There is a proof of length $\tilde{O}(K \cdot d)$ the **Cloud** can send, along with values $C(a_1), \dots, C(a_K) \in F$, such that
- the **Verifier** can toss $poly\left(\log\left(dK\frac{|F|}{\epsilon}\right)\right)$ coins and ***check the proof*** in about $\tilde{O}(K \cdot (n + d) + s)$ time, with probability of error $\leq \epsilon$.

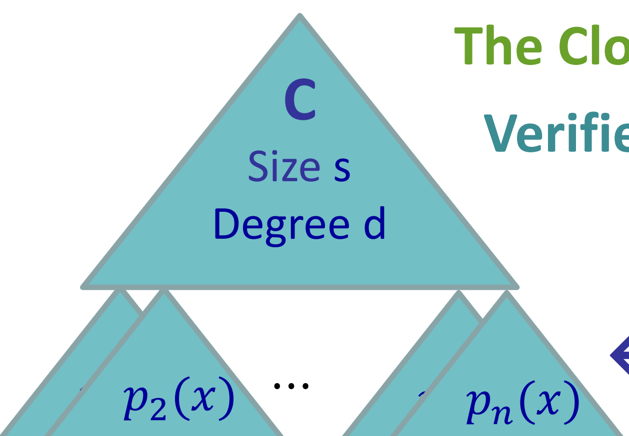
Given: $a_1, \dots, a_K \in F^n$
Want: $C(a_1), \dots, C(a_K) \in F$

Naively takes
~ $s \cdot K$ time… get ~ $s + K$





Idea: Define a *univariate* polynomial that simulates evaluation of C on the a_i



The Cloud sends *this* polynomial $Q(x)$, of degree $K \cdot d$.

Verifier: **1. Evals $Q(\beta_1), \dots, Q(\beta_K)$ to get $C(a_1), \dots, C(a_K)$**
2. Picks random $r \in F'$, F' large extension of F
3. Checks $Q(r) = C(p_1(r), \dots, p_n(r))$.

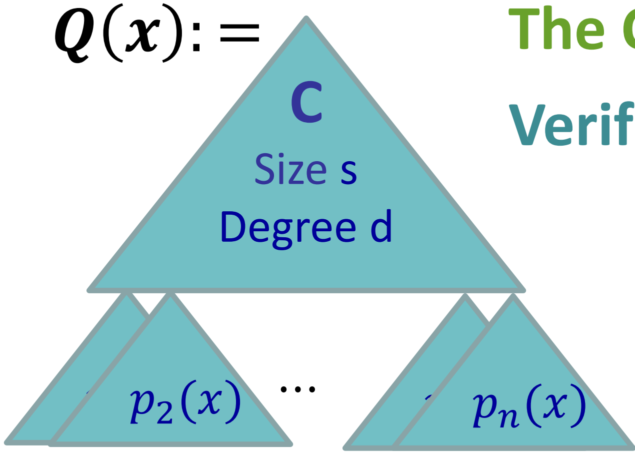
← *degree K polys.* **Let $\beta_1, \dots, \beta_K \in F$ be distinct.**
for all $i = 1, \dots, n$, & $j = 1, \dots, K$, $p_i(\beta_j) := a_j[i]$

Given: $a_1, \ldots, a_K \in F^n$
Want: $C(a_1), \ldots, C(a_K) \in F$

Would naively take $\approx s \cdot K$ time to compute

Idea: Define a *univariate* polynomial that simulates evaluation of C on the a_i

$Q(x) :=$



degree K polynomials

For all i, j, $p_i(\beta_j) := \alpha_j[i]$
for distinct $\beta_1, \ldots, \beta_K$

The Cloud sends $Q'(x)$, of degree $K \cdot d$

Verifier: 1. Evaluates $Q'(\beta_1), \ldots, Q'(\beta_K)$ to obtain the values $C(a_1), \ldots, C(a_K)$

[Use Fast Fourier Transform: $\widetilde{O}(K \cdot d)$ time]

2. Picks unifom random $r \in F'$, where

$$|F'| \geq poly\left(dK\frac{|F|}{\epsilon}\right)$$

F' is an extension field of F

3. Checks $Q'(r) = C(p_1(r), \ldots, p_n(r))$

[Evaluate each p_i on r, then evaluate C… Takes $\widetilde{O}(s + K \cdot n)$ time]

Correctness: [Euler??] For every pair $p(x), q(x)$ of $distinct$ polynomials of degree $\leq D$, $p(r) = q(r)$ for $\leq D + 1$ points r.

Thm: There is a proof system for UNSAT such that ***every UNSAT Boolean formula of poly(n) size*** has a proof of length $\sim 2^{n/2}$ verifiable with ***poly(n) bits of randomness*** in $\sim 2^{n/2}$ time.

Proof: **Let F be a field of char $\geq 2^n$. Given a Boolean formula B:**

1. Arithmetize B: WLOG, B has $O(\log n)$ depth.

Define n-variate polynomial P over F equivalent to B (over $\{0,1\}^n$) by replacing $(a \wedge b) \mapsto a \cdot b$, $(a \vee b) \mapsto a + b - a \cdot b$, $\neg a \mapsto (1 - a)$

$$\text{For all } a \in \{0,1\}^n,\ P(a_1, \ldots, a_n) = B(a_1, \ldots, a_n)$$

2. Partial-Sum P: Define a new polynomial P' in $n/2$ variables:

$$P'(x_1, \ldots, x_{n/2}) := \sum_{a_{n/2+1}, \ldots, a_n \in \{0,1\}} P(x_1, \ldots, x_{n/2}, a_{n/2+1}, \ldots, a_n)$$

Note that degree$(P') \leq poly(n)$ and size$(P') \leq 2^{n/2} \cdot poly(n)$

3. Apply Protocol! Cloud proves that $P'(a) = 0$ for all $a \in \{0,1\}^{n/2}$

The proof is a polynomial $Q(y)$ of degree $\leq poly(n) \cdot 2^{n/2}$ over F such that $Q(\beta_i) = P'(a_i)$, where $a_i \in \{0,1\}^{n/2}$.

Verifier checks $Q(\beta_i) = 0$ on all $2^{n/2}$ relevant points.

Conclusion

- Similar results hold for **Permanent**, **Counting Cliques**, …., **any** problem in the class **VNP**
- **3-Round (AMA) proof system for QBF in $2^{3n/4}$ time**
- ***Are there more tombstones?***

- Suppose we are given two arithmetic circuits $C(x), D(x)$ of size n, degree $\leq n$, and ***one*** variable x.

  Testing whether $C \equiv D$ is easy:
  - $\tilde{O}(n^2)$ deterministic time
  - $\tilde{O}(n)$ randomized time

Can $C \equiv D$ be tested in $\tilde{O}(n^{1.999})$ time?

A yes answer ➔ NSETH is false!

Thank you!