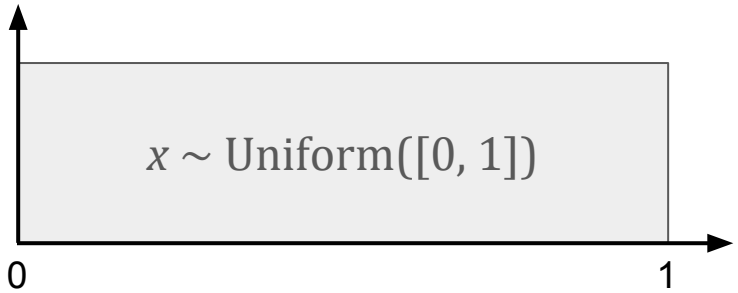
Efficient Synthesis with Probabilistic Constraints

Samuel Drews, Aws Albarghouthi, Loris D'Antoni

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

madPL

Probabilistic Correctness Properties



$post := \Pr[P(x) = 1] > ⅓$

$P(x) := 0 \leq x \leq 0.2$

✗

$P(x) := 0 \leq x \leq 0.8$

✓

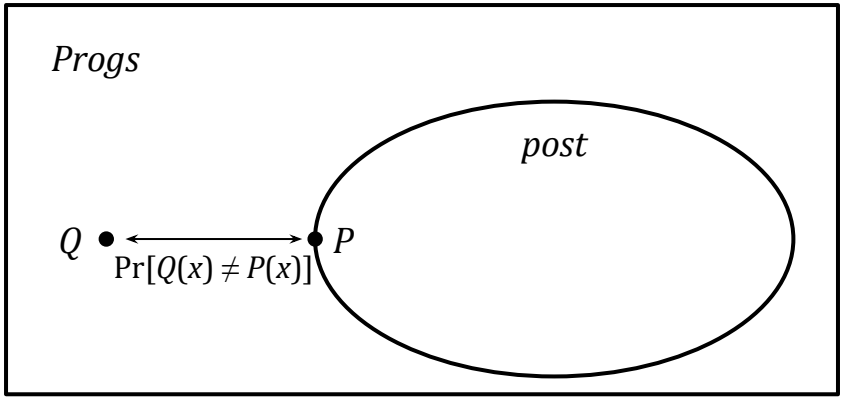
Probabilistic Program Synthesis

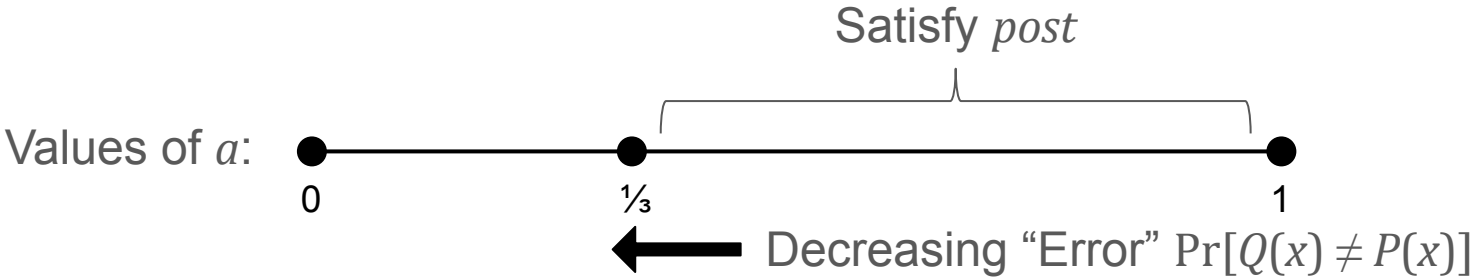
$Progs = \{P(x) := 0 \leq x \leq a \mid a \in \mathrm{R}\}$

$\mathrm{inputs} \sim \mathrm{Uniform}([0, 1])$

$post := \Pr[P(x) = 1] > ⅓$

$Q(x) :=$ constant-0 function





DIGITS (CAV17)

$Progs = \{P(x) := 0 \leq x \leq a \mid a \in \mathrm{R}\}$

$\mathrm{inputs} \sim \mathrm{Uniform}([0, 1])$

$post := \Pr[P(x) = 1] > ⅓$

$Q(x) :=$ constant-0 function

Sample m inputs

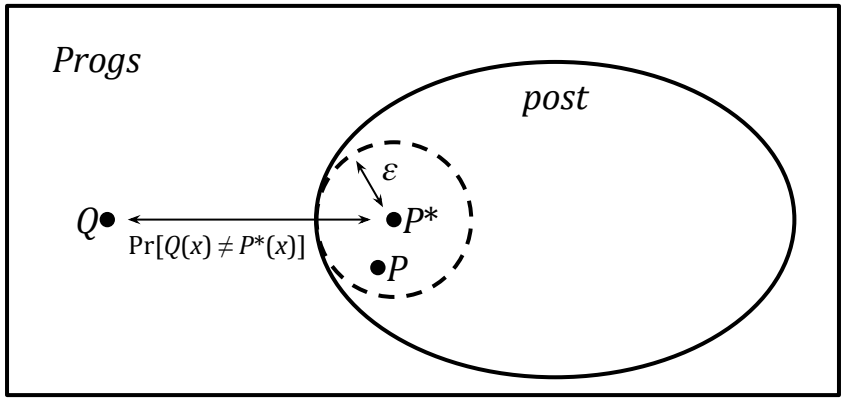
For each $f : S \to \{0,1\}$, query a synthesis oracle

Constraints $f : S \to \{0,1\}$

Sample Set S {0.4,	0.6}	Program	Correct?	Error
0	0	[0, 0.3]	✗	0.3
0	1	unsat	-	-
1	0	[0, 0.5]	✓	0.5
1	1	[0, 1]	✓	1.0

Oracles compute $post$ and Error

DIGITS Convergence (CAV17)

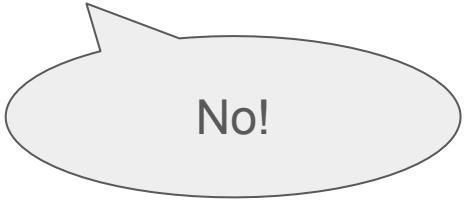


Theorem (Convergence of DIGITS)

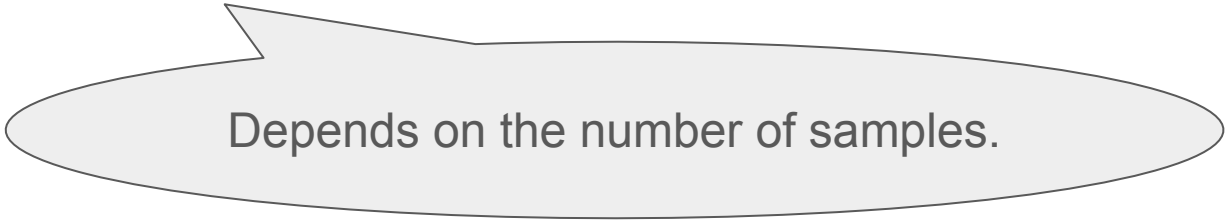
(Under certain assumptions,) with **high probability**, DIGITS enumerates some **correct, (near-)near-optimal** program P.

Which Begs the Questions:

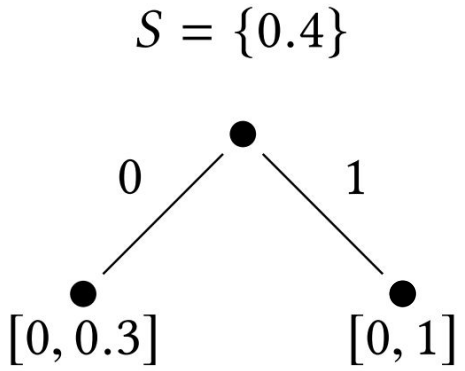
Do we really need to make exponentially many queries?

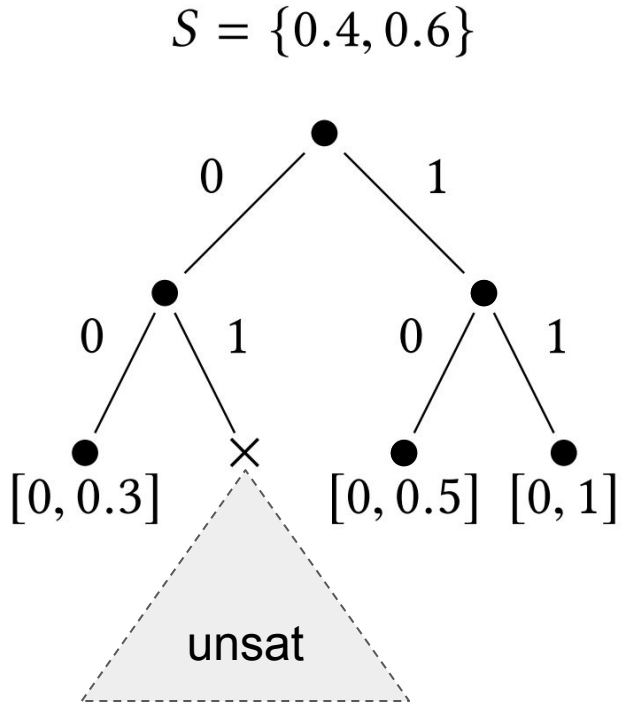


How high is the “high probability” of optimal convergence?



DIGITS Trie Implementation (CAV17)

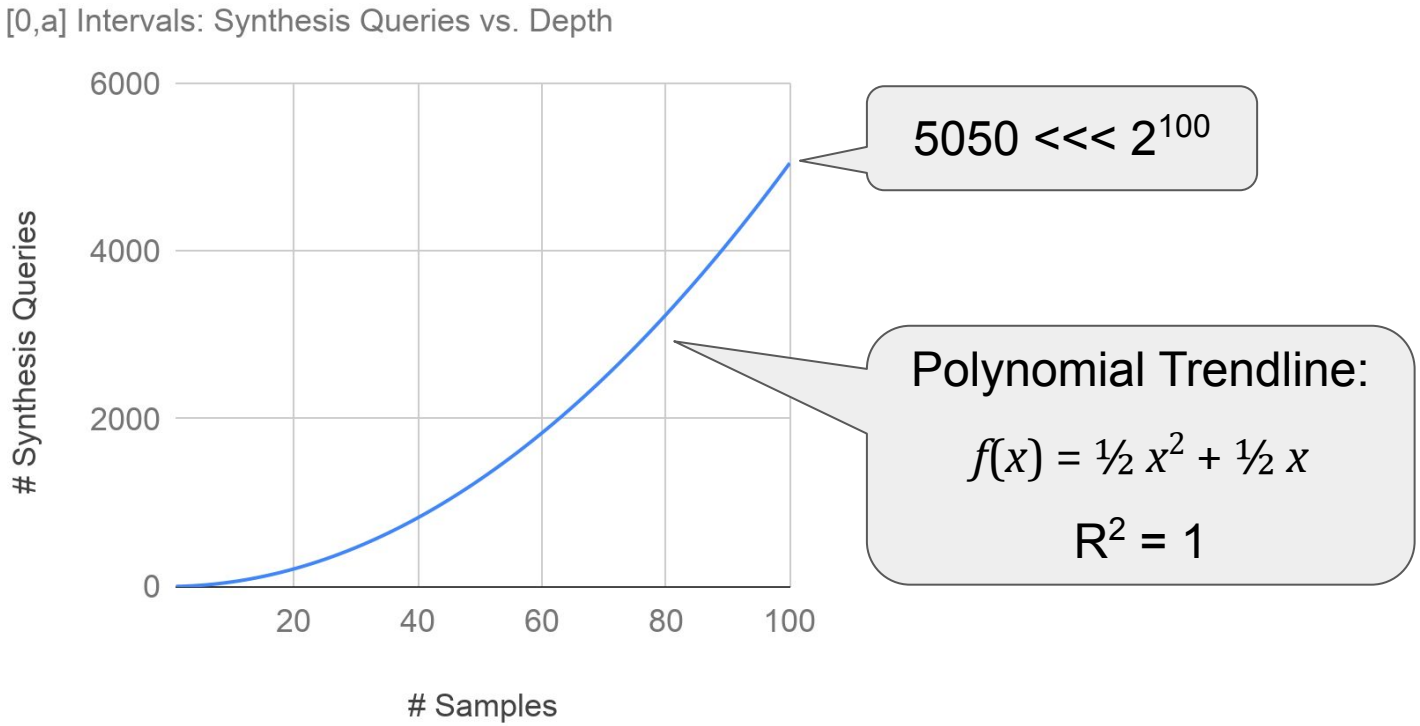




Constraints $f : S \to \{0,1\}$

Sample Set S {0.4,	0.6}	Program	Correct?	Error
0	0	[0, 0.3]	✗	0.3
0	1	unsat	-	-
1	0	[0, 0.5]	✓	0.5
1	1	[0, 1]	✓	1.0

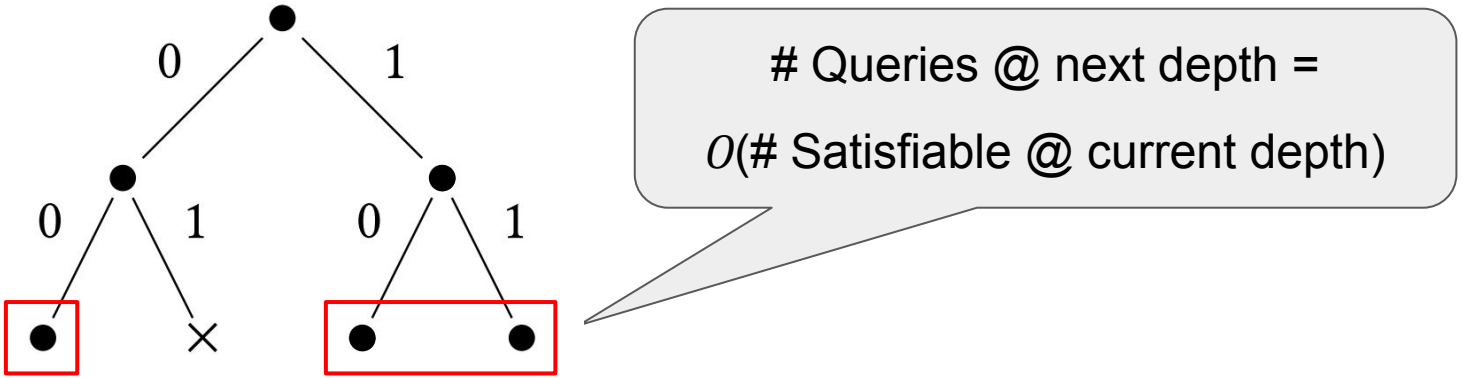
Efficiency of the Trie “Heuristic”



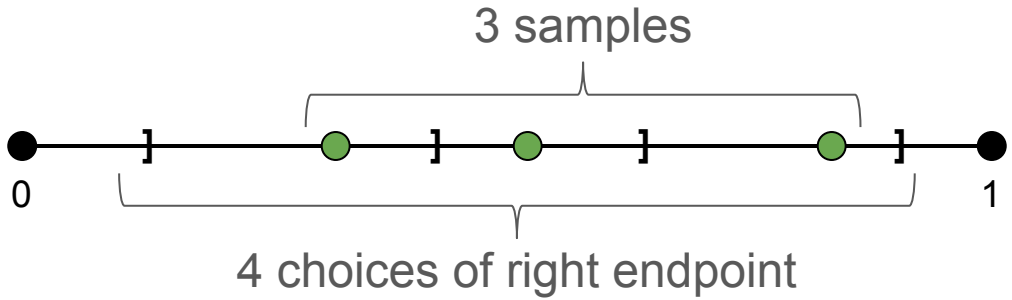
Polynomial?

Two equivalent ways of thinking:

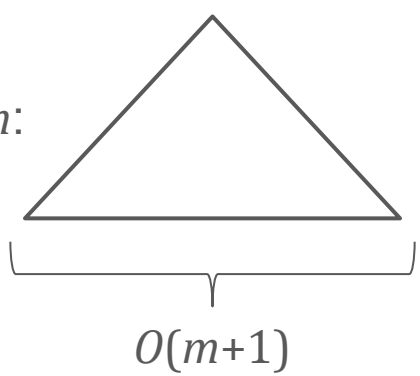
- The trie **avoids unsatisfiable** parts of the search space
- The trie **stays close to satisfiable** parts of the search space



For [0, a] intervals and m points, # Satisfiable Queries (m) = $m + 1$

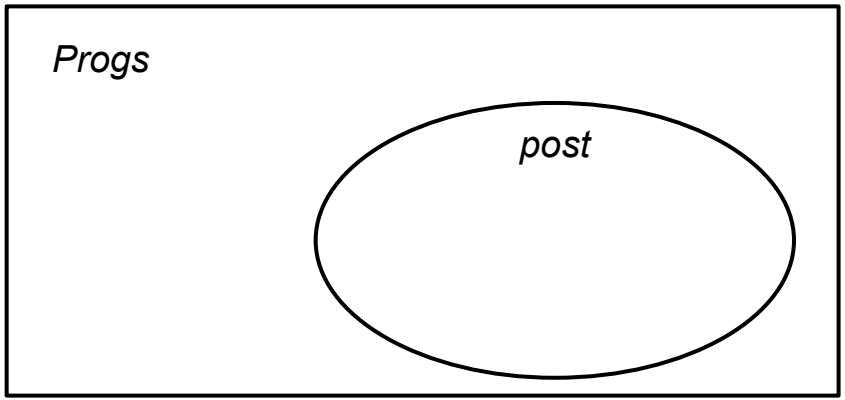


[0, a] Search Trie, depth m:

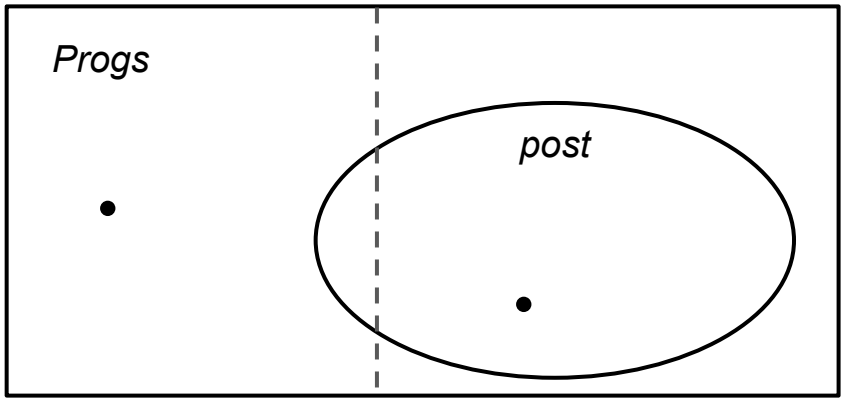


$\rightarrow$ $O(m^2)$

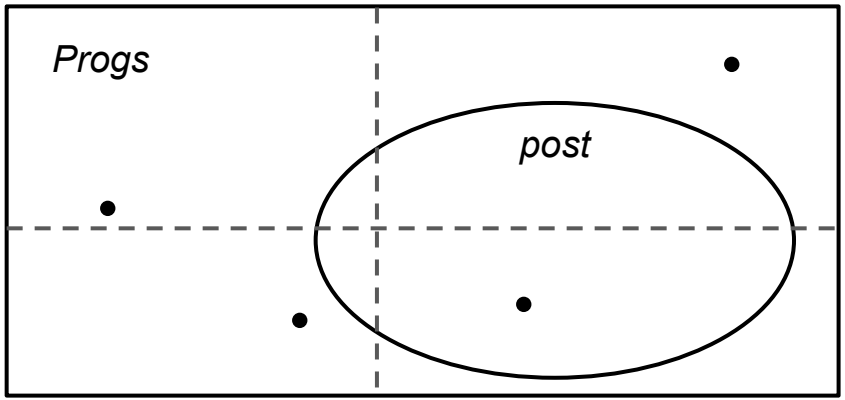
Finite VC Dimension → Polynomial Bound



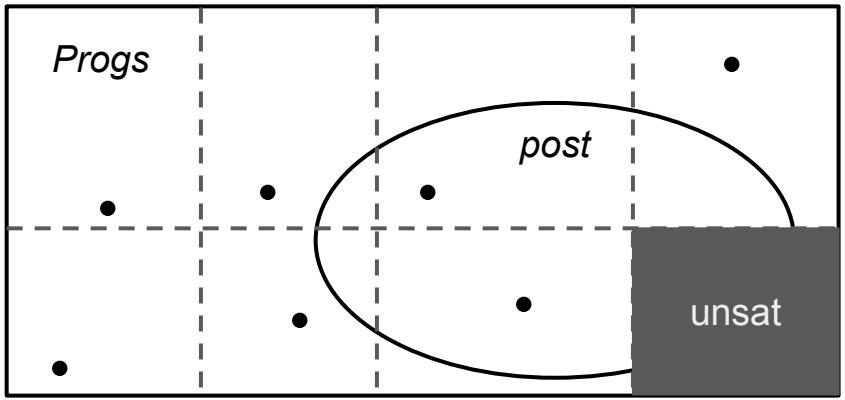
Finite VC Dimension → Polynomial Bound



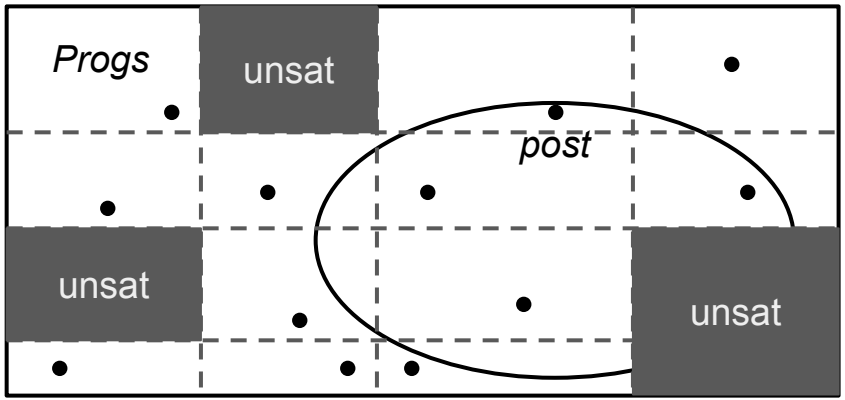
Finite VC Dimension → Polynomial Bound



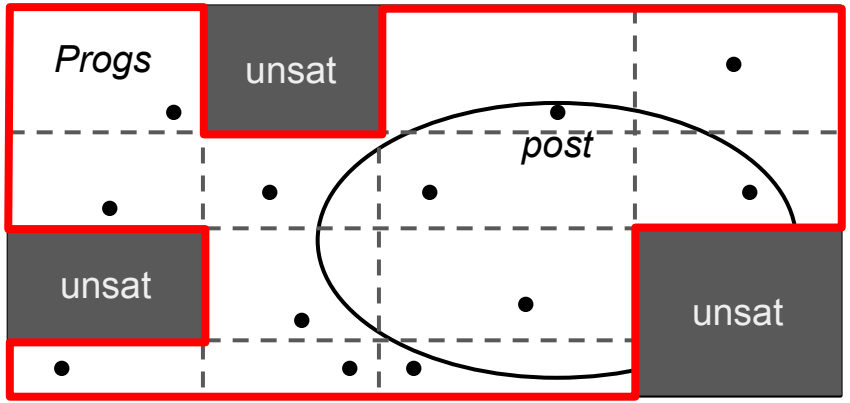
Finite VC Dimension → Polynomial Bound



Finite VC Dimension → Polynomial Bound



Finite VC Dimension → Polynomial Bound



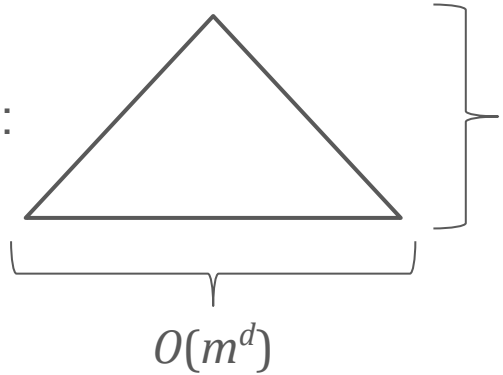
"Sauer-Shelah Lemma": Only $O(m^d)$ are satisfiable!

Finite VC Dimension → Polynomial Bound

Theorem:

If DIGITS runs on $Progs$ using m samples, it performs $O(m^{d+1})$ synthesis queries.

$Progs$ Search Trie, $|S| = m$:



→ $O(m^{d+1})$

Which Begs the Questions:

Do we really need to make exponentially many queries?

No!---In fact, we only make $O(m^{d+1})$

How high is the “high probability” of optimal convergence?

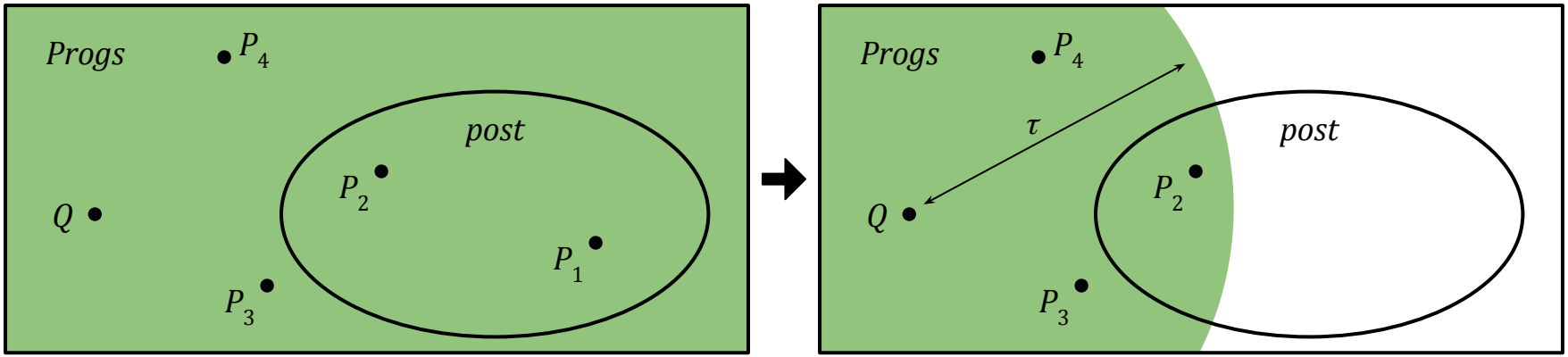
Depends on the number of samples.

τ-DIGITS

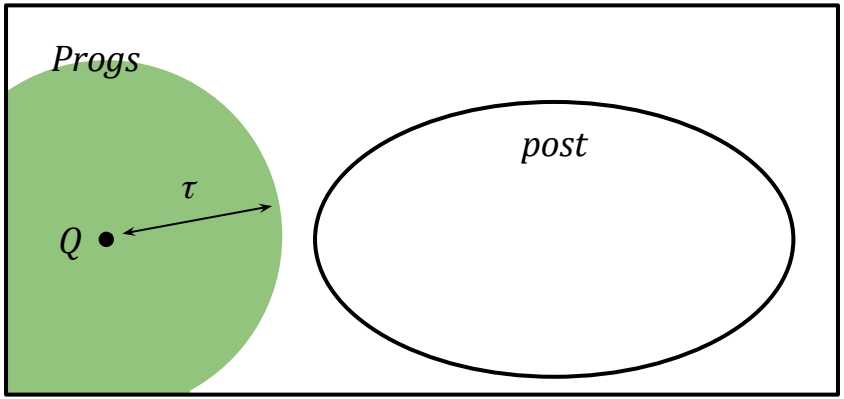
Use the constraint set to **estimate** the error $\Pr[Q(x) \neq P(x)]$:

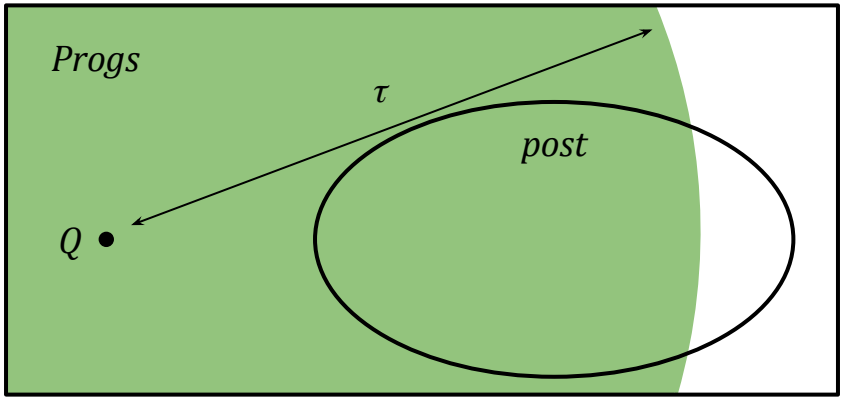
Fix threshold $\tau \in (0,1]$ and **skip** any $f: S \to \{0,1\}$ such that

$$|\{x \in S : Q(x) \neq f(x)\}| > \tau|S|$$

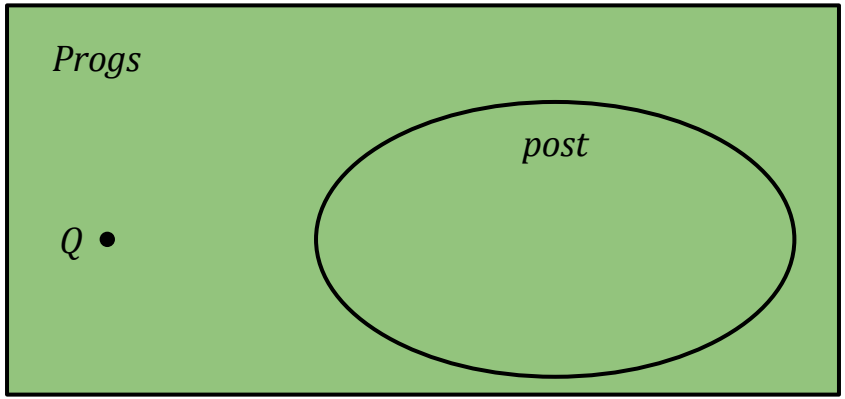


Choosing τ

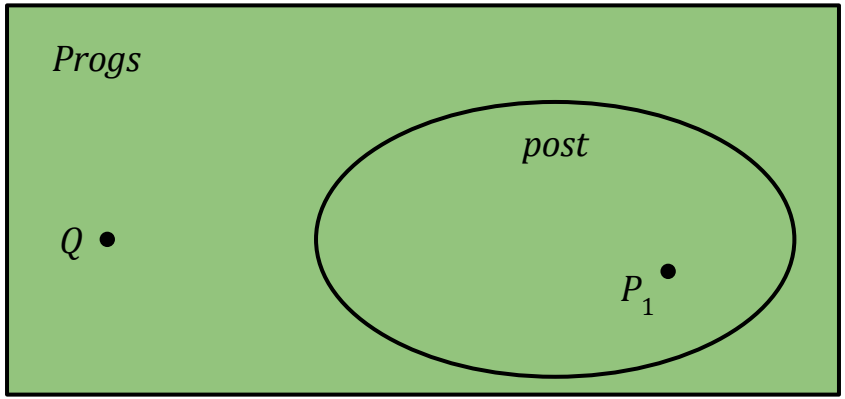




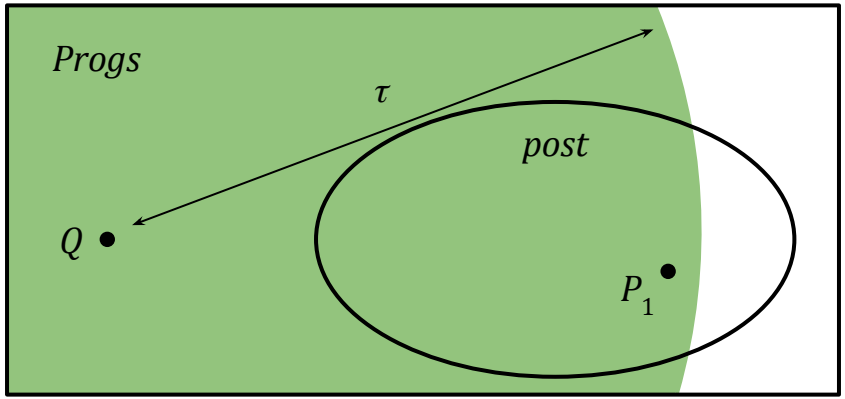
Adaptive τ-DIGITS



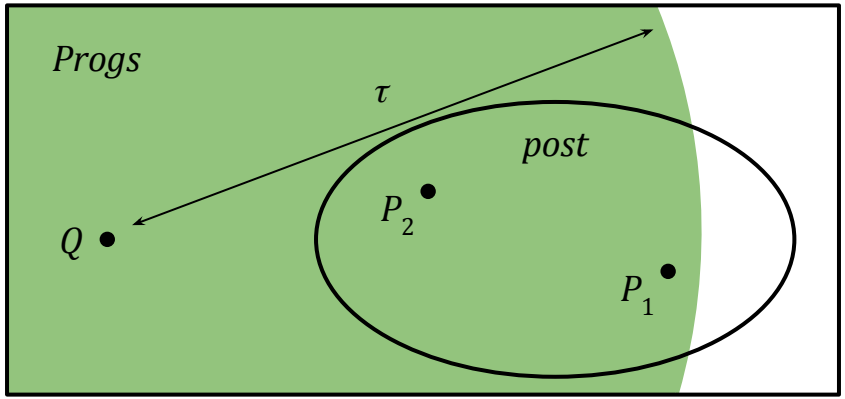
Adaptive τ-DIGITS



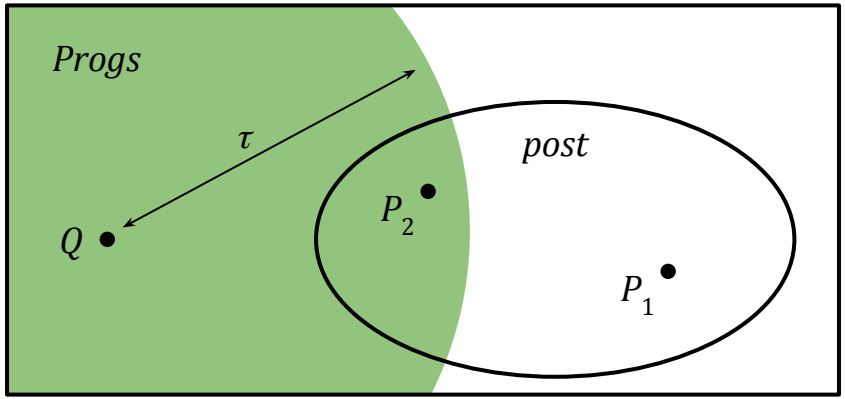
Adaptive τ-DIGITS



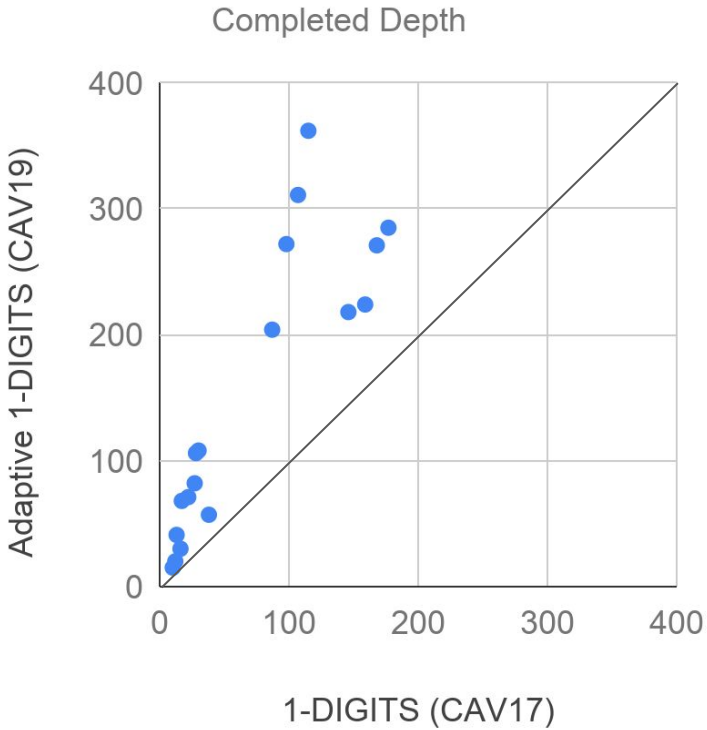
Adaptive τ-DIGITS

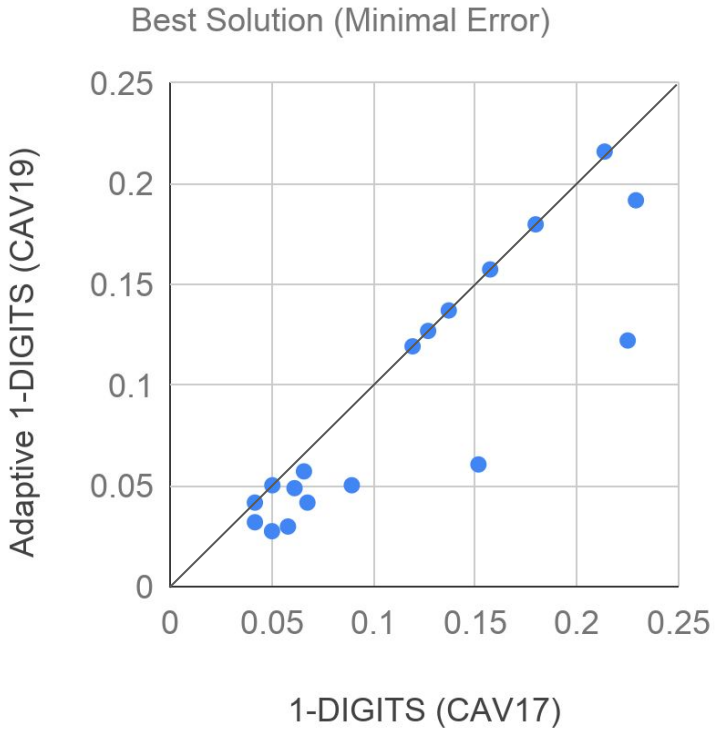


Adaptive τ-DIGITS



Adaptive τ-DIGITS Efficacy (Fairness Benchmarks)





Which Begs the Questions:

Do we really need to make exponentially many queries?

No!---in fact, we only make $O(m^{d+1})$

How high is the “high probability” of optimal convergence?

Depends on the number of samples.
---and we can do better with τ-DIGITS

Thanks!

Progs Search Trie, $|S| = m$: