A structural theorem for local algorithms with applications to coding, testing and privacy

Marcel Dall'Agnol
University of Warwick

Tom Gur
University of Warwick

Oded Lachish
Birkbeck, University of London

HALG 2022

Main result

Techniques

Applications

Main result

Techniques

Applications

An efficient transformation from local to sample-based algorithms.

An efficient transformation from local to sample-based algorithms.

Local: inspects a small part of the input string x
Sample-based: access by random (i, x_i) samples

An efficient transformation from local to sample-based algorithms.

Local: inspects a small part of the input string x
Sample-based: access by random (i, x_i) samples

But why?

Sample-based access buys

- Privacy

Sample-based access buys

- Privacy
- Efficient repetition: running q-query $L_1, \ldots, L_t$ on the same input takes
  - $O(qt \log t)$ queries in general
  - $O(q \log t)$ queries by **reusing samples**

Local + robust ($O(1)$ queries)		Local + sample-based ($o(n)$ queries)
L	$\Longrightarrow$	S

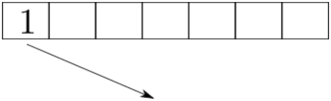
Local $+$ robust
($O(1)$ queries)

Local $+$ sample-based
($o(n)$ queries)

$$L \quad \Longrightarrow \quad S$$

Local + robust
($O(1)$ queries)

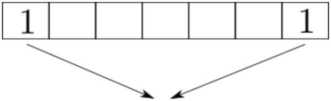
Local + sample-based
($o(n)$ queries)



$L \implies S$

Local $+$ robust ($O(1)$ queries)

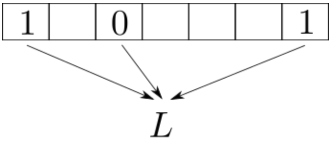
Local $+$ sample-based ($o(n)$ queries)



$$L \quad \Longrightarrow \quad S$$

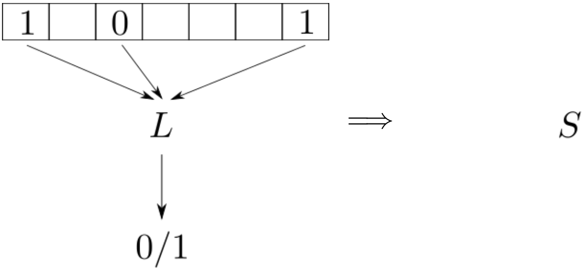
Local + robust
($O(1)$ queries)

Local + sample-based
($o(n)$ queries)



$\Longrightarrow$ S

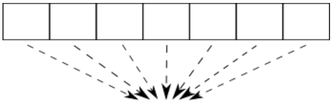
Local + robust ($O(1)$ queries) | Local + sample-based ($o(n)$ queries)



Local + robust ($O(1)$ queries)		Local + sample-based ($o(n)$ queries)
L	$\Longrightarrow$	S

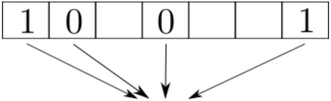
Local + robust
($O(1)$ queries)

Local + sample-based
($o(n)$ queries)



$$L \quad \Longrightarrow \quad S$$

Local + robust		Local + sample-based
($O(1)$ queries)		($o(n)$ queries)

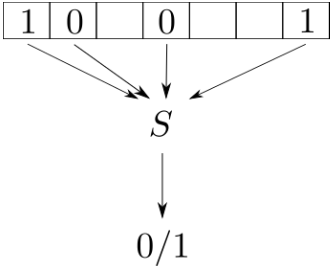


$$L \quad \Longrightarrow \quad S$$

Local + robust
($O(1)$ queries)

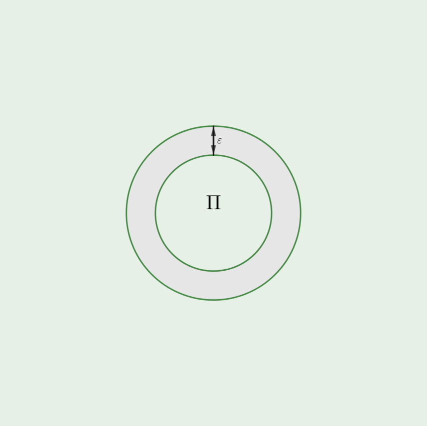
Local + sample-based
($o(n)$ queries)

$L \quad \Longrightarrow$



Property tester

Property $\Pi \subseteq \{0,1\}^n$, proximity parameter $\varepsilon > 0$

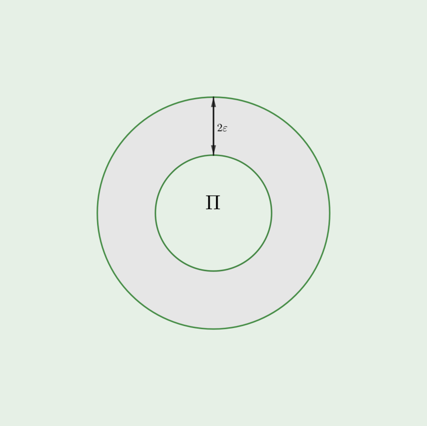


Tester T computes

$$f(x) = \begin{cases} 1, & \text{if } x \in \Pi \\ 0, & \text{if } x \text{ is } \varepsilon\text{-far from } \Pi. \end{cases}$$

Property tester

Property $\Pi \subseteq \{0,1\}^n$, proximity parameter $\varepsilon > 0$

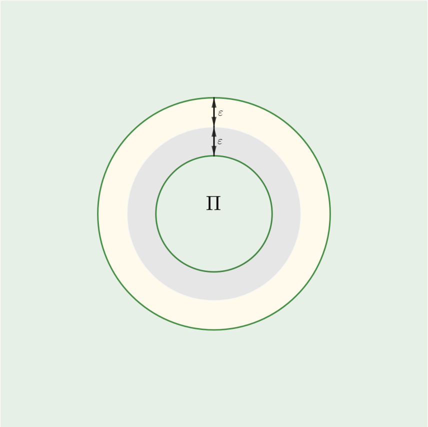


Tester T computes

$$f(x) = \begin{cases} 1, & \text{if } x \in \Pi \\ 0, & \text{if } x \text{ is } \varepsilon\text{-far from } \Pi. \end{cases}$$

Property tester

Property $\Pi \subseteq \{0,1\}^n$, proximity parameter $\varepsilon > 0$

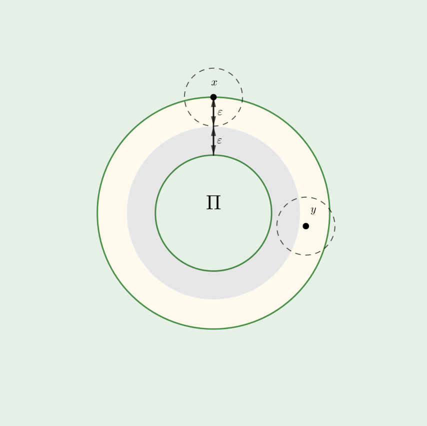


Tester T **robustly** computes

$$g(x) = \begin{cases} 1, & \text{if } x \in \Pi \\ 0, & \text{if } x \text{ is } 2\varepsilon\text{-far from } \Pi. \end{cases}$$

Property tester

Property $\Pi \subseteq \{0,1\}^n$, proximity parameter $\varepsilon > 0$



Tester T **robustly** computes

$$g(x) = \begin{cases} 1, & \text{if } x \in \Pi \\ 0, & \text{if } x \text{ is } 2\varepsilon\text{-far from } \Pi. \end{cases}$$

Theorem

Any function computed by a q-query robust local algorithm admits a sample-based algorithm with sample complexity

$$n^{1-\tilde{\Omega}(1/q^2)}.$$

$$(q = \Omega(\sqrt{\log n}) \implies \text{sample complexity } \Omega(n))$$

Theorem

Any function computed by a q-query robust local algorithm admits a sample-based algorithm with sample complexity

$$n^{1-\tilde{\Omega}(1/q^2)}.$$

$$(q = \Omega(\sqrt{\log n}) \implies \text{sample complexity } \Omega(n))$$

Theorem

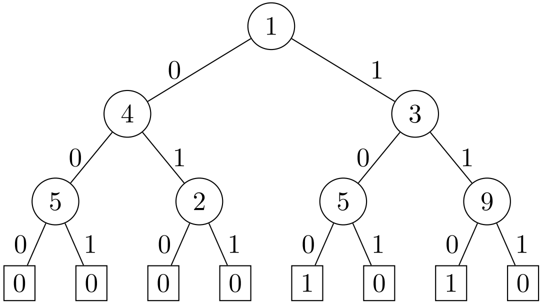
This transformation cannot achieve sample complexity

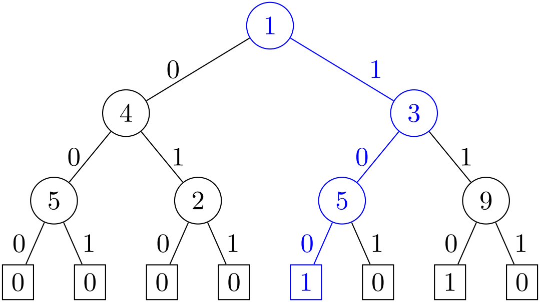
$$n^{1-\omega(1/q)}.$$

Main result

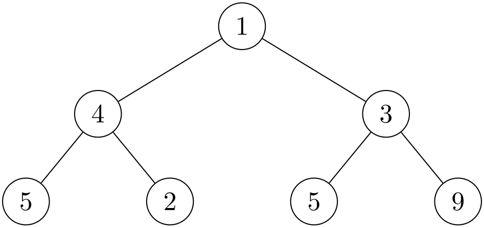
Techniques

Applications

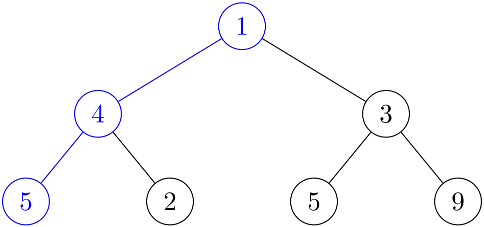

1
0
1
4
3
0
1
0
1
5
2
5
9
0
1
0
1
0
1
0
1
0
0
0
0
1
0
1
0



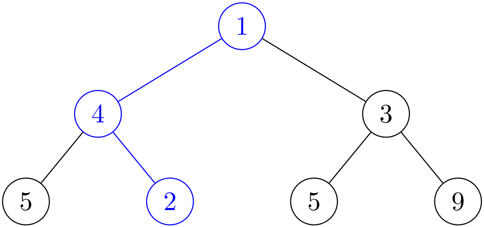
$$x_1 = 1$$
$$x_3 = 0$$
$$x_5 = 0$$



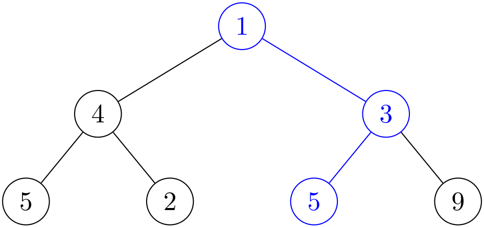
$$\mathcal{Q}_T = \left\{ \left\{ \begin{matrix} 1 \\ 4 \\ 5 \end{matrix} \right\}, \left\{ \begin{matrix} 1 \\ 4 \\ 2 \end{matrix} \right\}, \left\{ \begin{matrix} 1 \\ 3 \\ 5 \end{matrix} \right\}, \left\{ \begin{matrix} 1 \\ 3 \\ 9 \end{matrix} \right\} \right\}$$



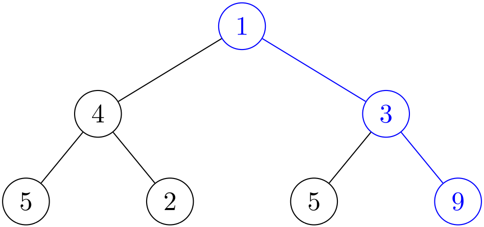
$$\mathcal{Q}_T = \left\{ \begin{Bmatrix} 1 \\ 4 \\ 5 \end{Bmatrix}, \begin{Bmatrix} 1 \\ 4 \\ 2 \end{Bmatrix}, \begin{Bmatrix} 1 \\ 3 \\ 5 \end{Bmatrix}, \begin{Bmatrix} 1 \\ 3 \\ 9 \end{Bmatrix} \right\}$$



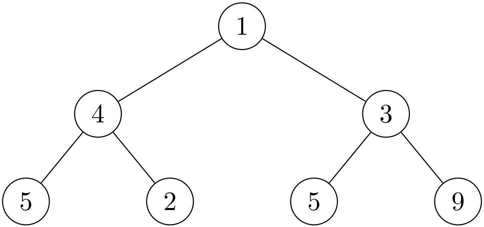
$$\mathcal{Q}_T = \left\{ \quad \left\{ \begin{array}{c} 1 \\ 4 \\ 5 \end{array} \right\}, \quad \left\{ \begin{array}{c} 1 \\ 4 \\ 2 \end{array} \right\}, \quad \left\{ \begin{array}{c} 1 \\ 3 \\ 5 \end{array} \right\}, \quad \left\{ \begin{array}{c} 1 \\ 3 \\ 9 \end{array} \right\} \quad \right\}$$



$$\mathcal{Q}_T = \left\{ \quad \left\{ \begin{array}{c} 1 \\ 4 \\ 5 \end{array} \right\}, \quad \left\{ \begin{array}{c} 1 \\ 4 \\ 2 \end{array} \right\}, \quad \left\{ \begin{array}{c} 1 \\ 3 \\ 5 \end{array} \right\}, \quad \left\{ \begin{array}{c} 1 \\ 3 \\ 9 \end{array} \right\} \quad \right\}$$



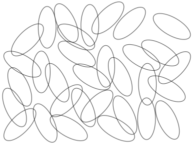
$$\mathcal{Q}_T = \left\{ \quad \left\{ \begin{matrix} 1 \\ 4 \\ 5 \end{matrix} \right\}, \quad \left\{ \begin{matrix} 1 \\ 4 \\ 2 \end{matrix} \right\}, \quad \left\{ \begin{matrix} 1 \\ 3 \\ 5 \end{matrix} \right\}, \quad \left\{ \begin{matrix} 1 \\ 3 \\ 9 \end{matrix} \right\} \quad \right\}$$



$$\mathcal{Q}_T = \left\{ \left\{ \begin{array}{c} 1 \\ 4 \\ 5 \end{array} \right\}, \quad \left\{ \begin{array}{c} 1 \\ 4 \\ 2 \end{array} \right\}, \quad \left\{ \begin{array}{c} 1 \\ 3 \\ 5 \end{array} \right\}, \quad \left\{ \begin{array}{c} 1 \\ 3 \\ 9 \end{array} \right\} \right\}$$

$$\mathcal{Q} = \bigcup_T \mathcal{Q}_T$$

Daisy partition theorem

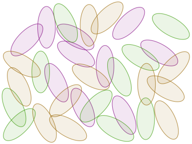


$\mathcal{Q}$

An *arbitrary* $\mathcal{Q}$ ($|\mathcal{Q}| \approx n$) can be partitioned into $\mathcal{D}_0, \ldots, \mathcal{D}_q$.
$\mathcal{D}_i$ has:

- petals of size i
- small kernel $|K_i|$
- bounded petal intersection

Daisy partition theorem

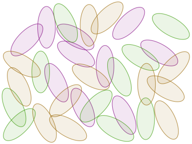


$\mathcal{Q}$

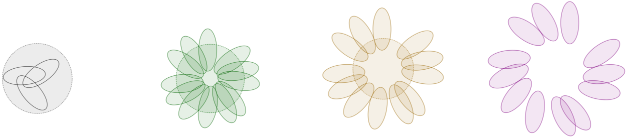
An *arbitrary* $\mathcal{Q}$ ($|\mathcal{Q}| \approx n$) can be partitioned into $\mathcal{D}_0, \ldots, \mathcal{D}_q$.
$\mathcal{D}_i$ has:

- petals of size i
- small kernel $|K_i|$
- bounded petal intersection

Daisy partition theorem



$\downarrow$

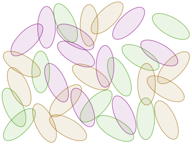


$\mathcal{D}_0$ $\mathcal{D}_1$ $\mathcal{D}_2$ $\mathcal{D}_3$

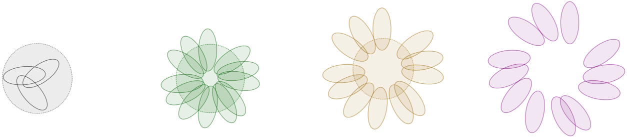
An *arbitrary* $\mathcal{Q}$ ($|\mathcal{Q}| \approx n$) can be partitioned into $\mathcal{D}_0, \ldots, \mathcal{D}_q$.
$\mathcal{D}_i$ has:

- petals of size i
- small kernel $|K_i|$
- bounded petal intersection

Daisy partition theorem



$\downarrow$



$\mathcal{D}_0$ $\mathcal{D}_1$ $\mathcal{D}_2$ $\mathcal{D}_3$

An *arbitrary* $\mathcal{Q}$ ($|\mathcal{Q}| \approx n$) can be partitioned into $\mathcal{D}_0, \ldots, \mathcal{D}_q$.
$\mathcal{D}_i$ has:

- petals of size i
- small kernel $|K_i|$
- bounded petal intersection

Main result

Techniques

Applications

Relaxed LDC lower bound

Any $C : \{0,1\}^k \to \{0,1\}^n$ has $n = k^{1+\tilde{\Omega}(1/q^2)}$.

Relaxed LDC lower bound

Any $C : \{0,1\}^k \to \{0,1\}^n$ has $n = k^{1+\tilde{\Omega}(1/q^2)}$.

(Previously $k^{1+\tilde{\Omega}\left(1/2^{2q}\right)}$; best upper bound is $k^{1+O(1/q)}$)

Relaxed LDC lower bound

Any $C : \{0,1\}^k \to \{0,1\}^n$ has $n = k^{1+\tilde{\Omega}(1/q^2)}$.

(Previously $k^{1+\tilde{\Omega}\left(1/2^{2q}\right)}$; best upper bound is $k^{1+O(1/q)}$)

Near-optimality of "P $\neq$ NP for testers"

q-query tester with short proof $\implies$ $n^{1-\tilde{\Omega}(1/q^2)}$-query tester.

Relaxed LDC lower bound

Any $C : \{0,1\}^k \to \{0,1\}^n$ has $n = k^{1+\tilde{\Omega}(1/q^2)}$.

(Previously $k^{1+\tilde{\Omega}\left(1/2^{2q}\right)}$; best upper bound is $k^{1+O(1/q)}$)

Near-optimality of "P $\neq$ NP for testers"

q-query tester with short proof $\implies$ $n^{1-\tilde{\Omega}(1/q^2)}$-query tester.

Best separation: q vs. $n^{1-O(1/q)}$ queries with $O(\log n)$-long proof

Relaxed LDC lower bound

Any $C : \{0,1\}^k \to \{0,1\}^n$ has $n = k^{1+\tilde{\Omega}(1/q^2)}$.

(Previously $k^{1+\tilde{\Omega}\left(1/2^{2q}\right)}$; best upper bound is $k^{1+O(1/q)}$)

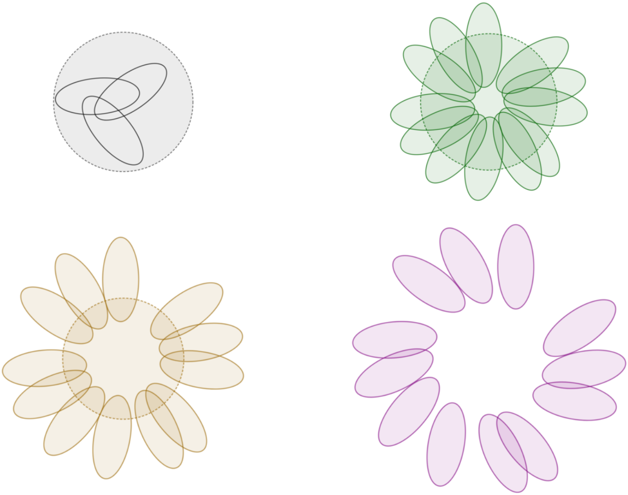
Near-optimality of "P $\neq$ NP for testers"

q-query tester with short proof $\implies$ $n^{1-\tilde{\Omega}(1/q^2)}$-query tester.

Best separation: q vs. $n^{1-O(1/q)}$ queries with $O(\log n)$-long proof

Adaptive-to-sample-based transformation for testers

q-query tester $\implies$ $n^{1-\tilde{\Omega}(1/q^2)}$-query sample-based tester.



Thank you!