Erasures vs. Errors in Property Testing and Local List Decoding

Sofya Raskhodnikova

Boston University

Joint work with Noga Ron-Zewi (*Haifa University*)
Nithin Varma (*Boston University*)

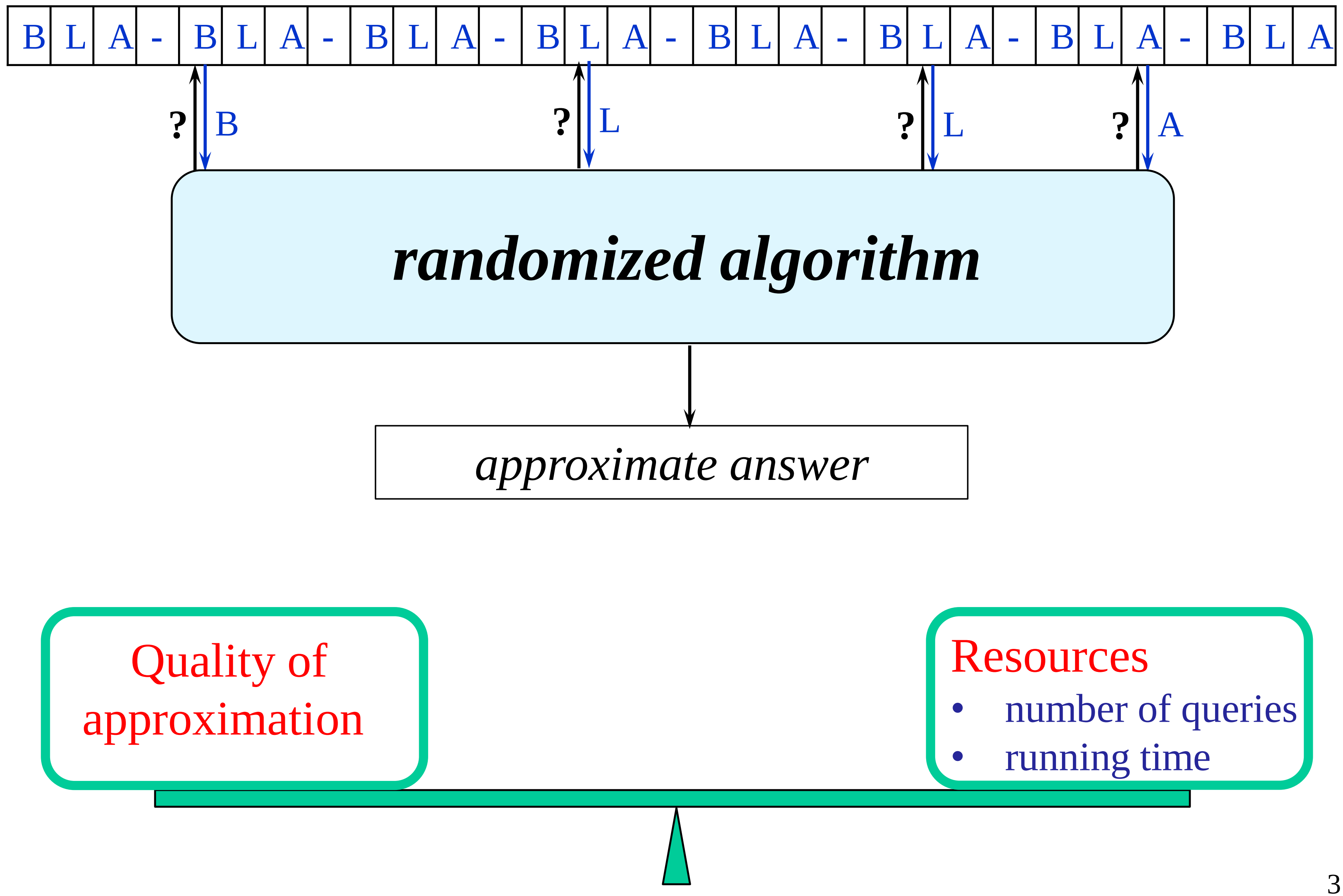
BOSTON UNIVERSITY

Goal: study of sublinear algorithms resilient to adversarial corruptions in the input

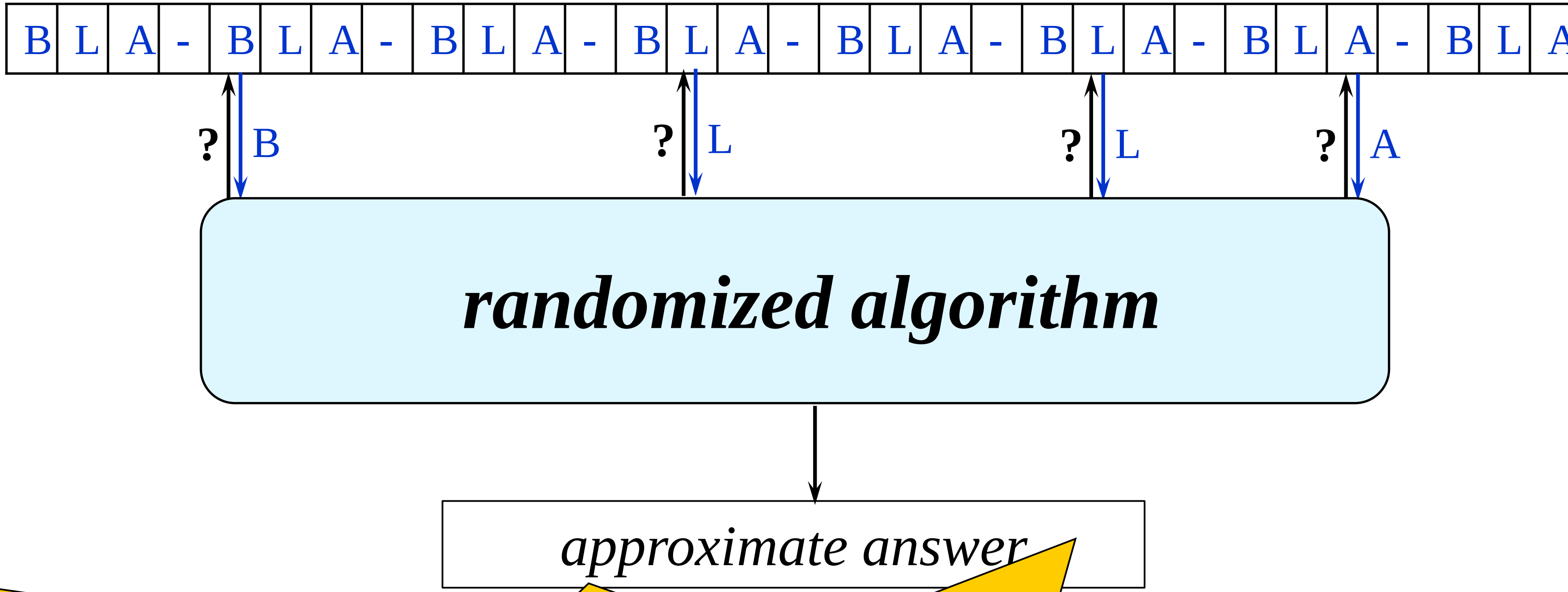
Focus: property testing model

[Rubinfeld Sudan 96, Goldreich Goldwasser Ron 98]

A Sublinear-Time Algorithm

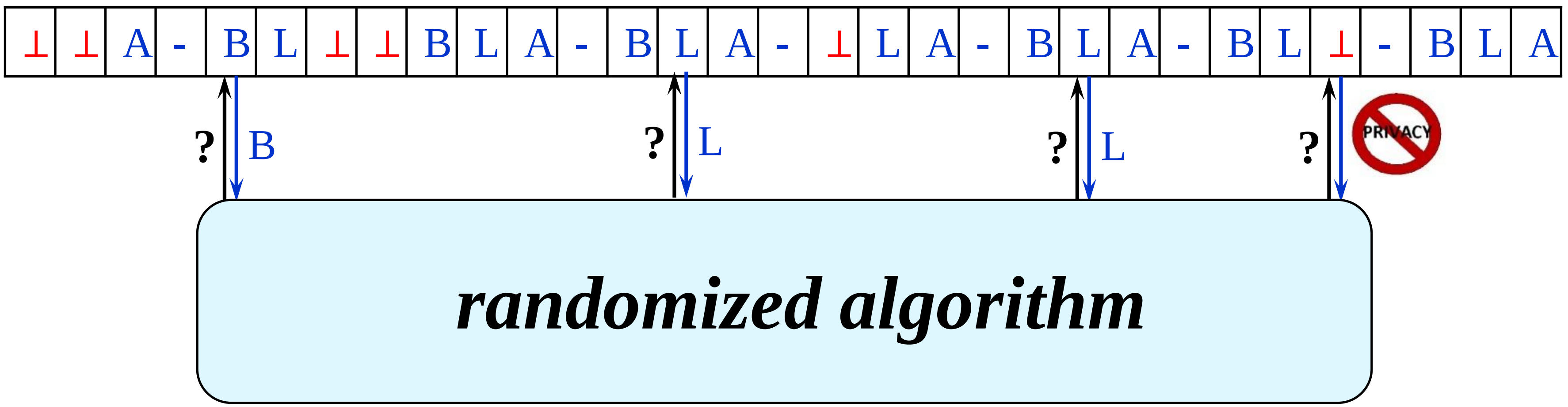


A Sublinear-Time Algorithm



Is it always reasonable to assume the input is intact?

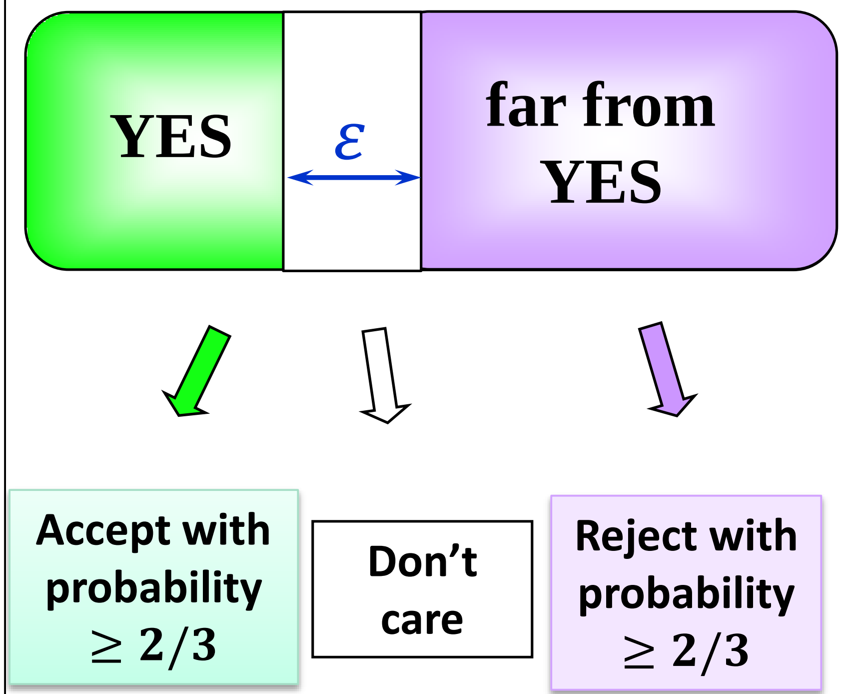
Algorithms Resilient to Erasures (or Errors)



- $\leq \boldsymbol{\alpha}$ fraction of the input is erased (or modified) adversarially before algorithm runs
- Algorithm does not know in advance what’s erased (or modified)
- Can we still perform computational tasks?

Property Testing

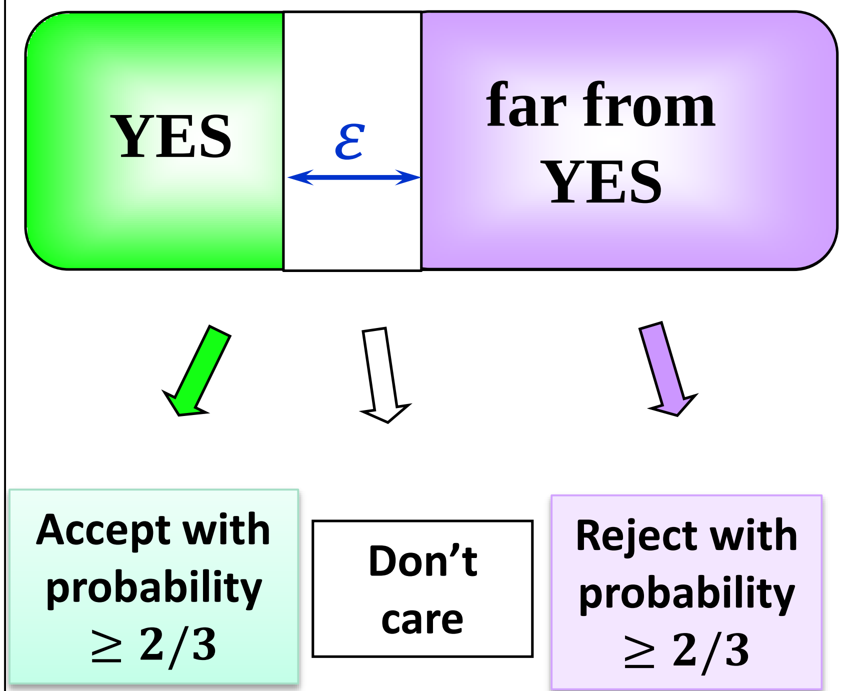
Property Tester [Rubinfeld Sudan 96, Goldreich Goldwasser Ron 98]



Two objects are at distance ε = they differ in an ε fraction of places

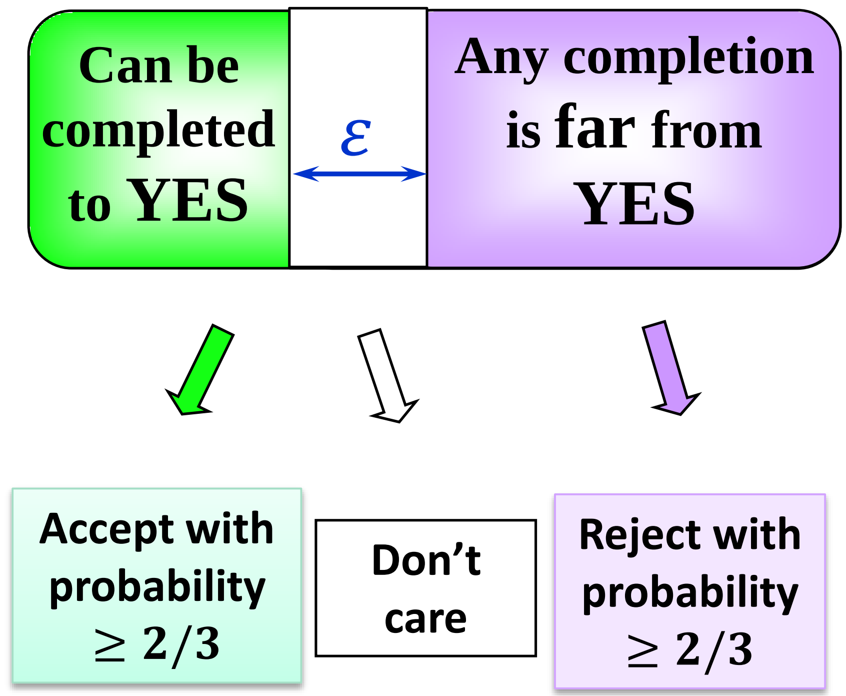
Property Testing with Erasures

Property Tester [Rubinfeld Sudan 96, Goldreich Goldwasser Ron 98]



Erasure-Resilient Property Tester [Dixit Raskhodnikova Thakurta Varma 16]

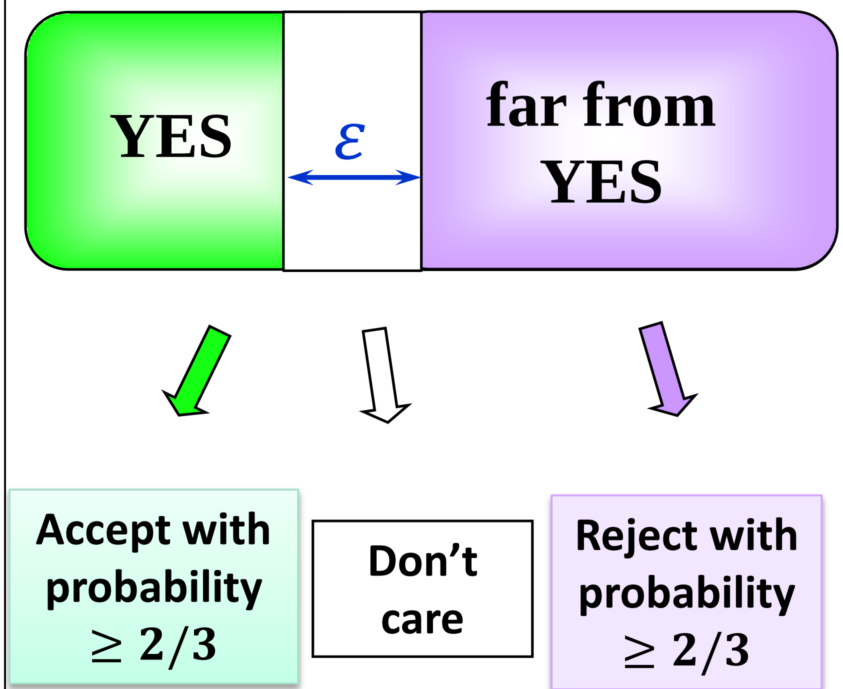
- $\leq \alpha$ fraction of the input is erased adversarially



Two objects are at distance ε = they differ in an ε fraction of places

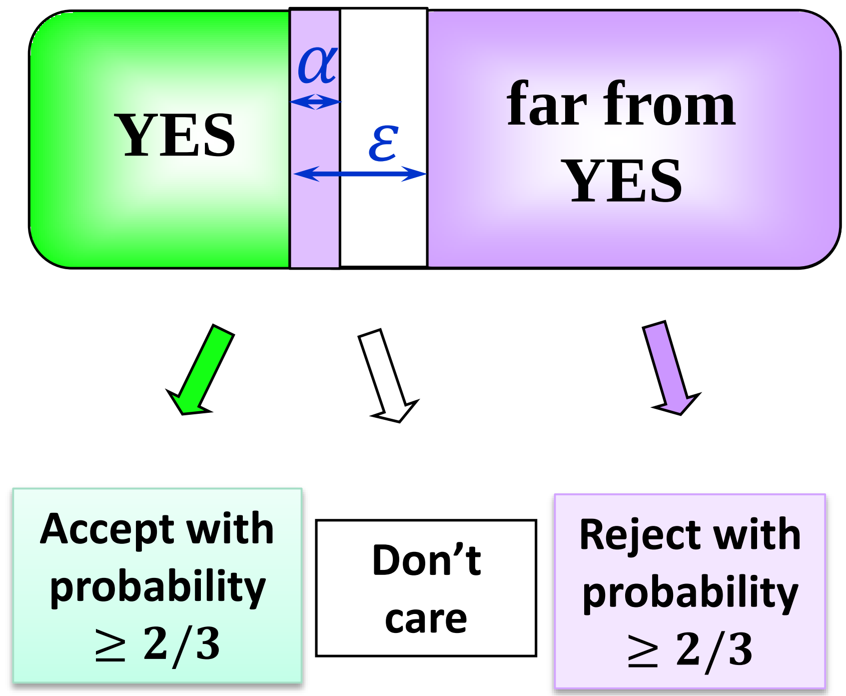
Property Testing with Errors

Property Tester [Rubinfeld Sudan 96, Goldreich Goldwasser Ron 98]



Tolerant Property Tester [Parnas Ron Rubinfeld 06]

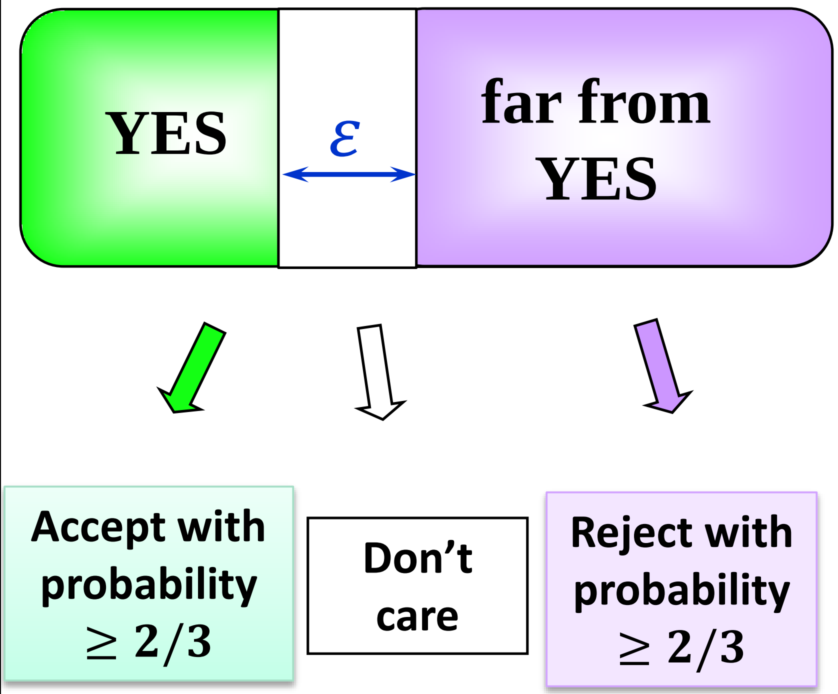
- $\leq \alpha$ fraction of the input is wrong



Two objects are at distance ε = they differ in an ε fraction of places

Property Testing with Errors

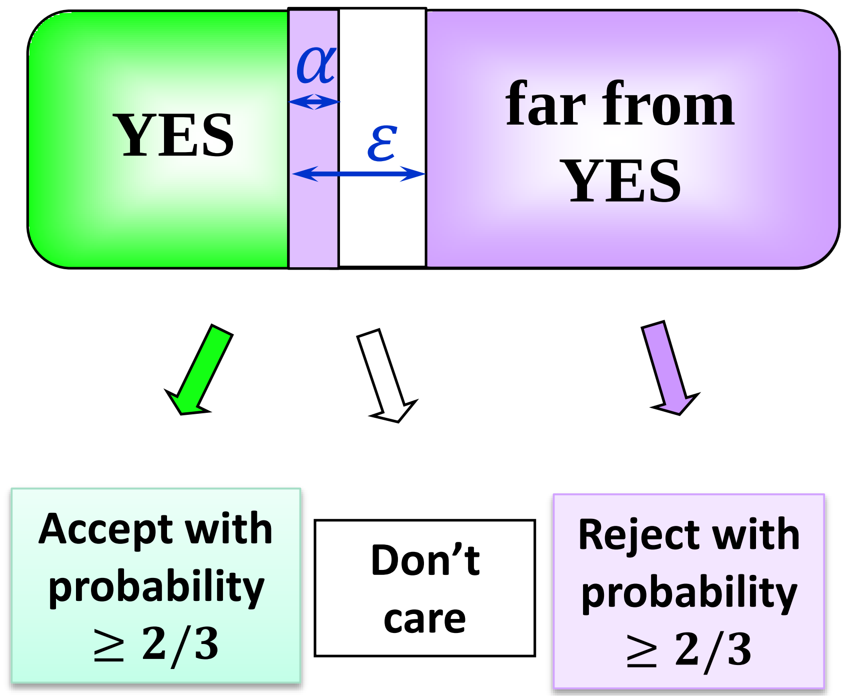
Property Tester [Rubinfeld Sudan 96, Goldreich Goldwasser Ron 98]



Tolerant Property Tester

[Parnas Ron Rubinfeld 06]

- $\leq \alpha$ fraction of the input is wrong

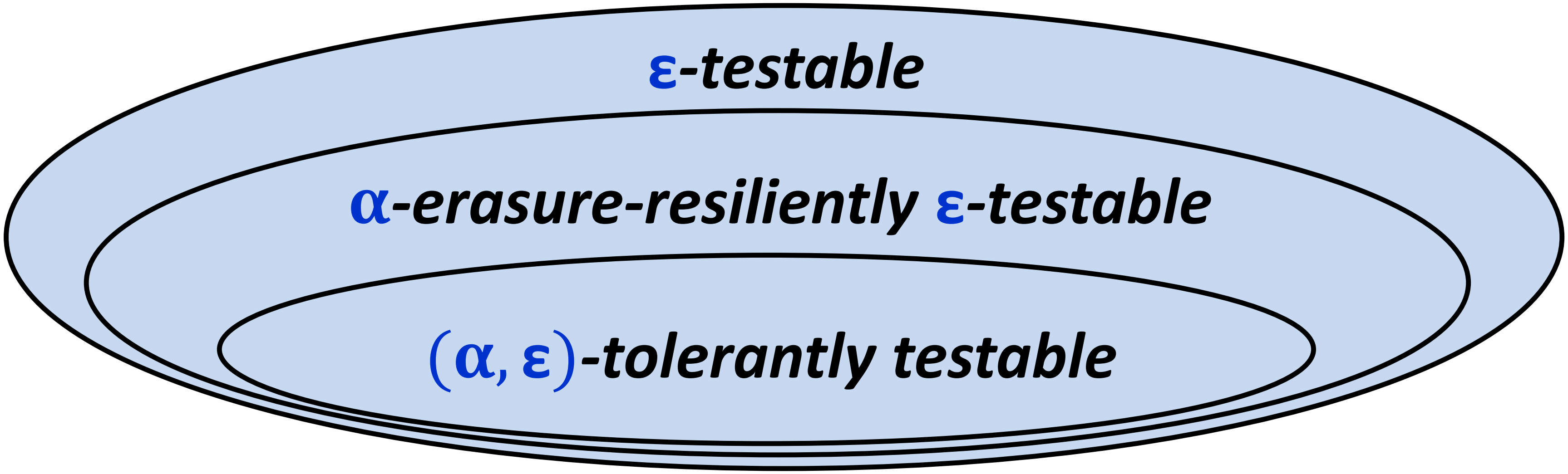


Two objects are at distance ε = they differ in an ε fraction of places

Relationships Between Models

Containments are strict:

- [Fischer Fortnow 05]: standard vs. tolerant
- [Dixit Raskhodnikova Thakurta Varma 16]: standard vs. erasure-resilient
- **new**: erasure-resilient vs. tolerant



Our Separation

Separation Theorem

There is a property of $\boldsymbol{n}$-bit strings that

- can be $\boldsymbol{\alpha}$-resiliently $\boldsymbol{\varepsilon}$-tested with **constant** query complexity,
- but requires $\boldsymbol{n}^{\boldsymbol{\Omega(1)}}$ queries for tolerant testing.

Most of the talk: **constant** vs. $\boldsymbol{\Omega(\log n)}$ separation.

Main Tool: Locally List Erasure-Decodeable Codes

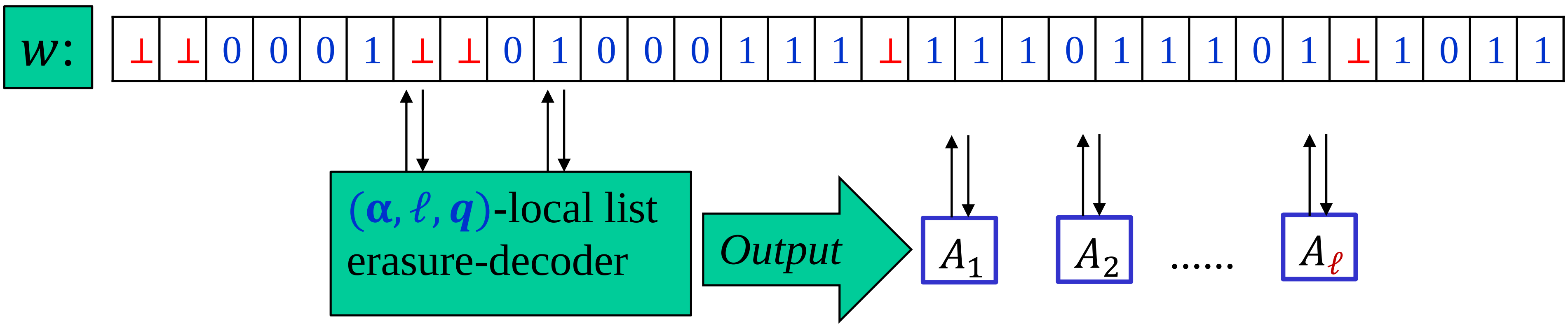
- Locally list decodable codes have been extensively studied [Goldreich Levin 89, Sudan Trevisan Vadhan 01, Gutfreund Rothblum 08, Gopalan Klivans Zuckerman 08, Ben-Aroya Efremenko Ta-Shma 10, Kopparty Saraf 13, Kopparty 15, Hemenway Ron-Zewi Wootters 17, Goi Kopparty Oliveira Ron-Zewi Saraf 17, Kopparty Ron-Zewi Saraf Wootters 18]

- Only errors, not erasures were previously considered
  - Not the case without the locality restriction [Guruswami 03, Guruswami Indyk 05]

Can locally list decodable codes perform better with erasures than with errors?

A Locally List Erasure-Decodable Code

- An error-correcting code $\boldsymbol{\mathcal{C}}_n\colon \Sigma^n \rightarrow \Sigma^N$
- Parameters: $\boldsymbol{\alpha}$ fraction of erasures, list size ℓ and $\boldsymbol{q}$ queries.



  – the fraction of erased bits in w is at most $\boldsymbol{\alpha}$,
  – the decoder makes at most $\boldsymbol{q}$ queries to w,
  – w.p. $\geq 2/3$, for every $x \in \Sigma^n$ with encoding $\boldsymbol{\mathcal{C}}_n(x)$ that agrees with w on all non-erased bits, one of the algorithms A_j, given oracle access to w, implicitly computes x (that is, $A_j(i) = x_i$);
  – each algorithm A_j makes at most $\boldsymbol{q}$ queries to w.

Hadamard Code

- Hadamard: $\{0,1\}^k \to \{0,1\}^{2^k}$; Hadamard$(x) = (\langle x, y\rangle)_{y\in\{0,1\}^k}$
- Impossible to decode when fraction of errors $\boldsymbol{\alpha} \geq \mathbf{1/2}$.

Type of corruptions	Corruption tolerance $\boldsymbol{\alpha}$	List size, ℓ	Number of queries, q	Upper bound	Lower bound
Errors	$\alpha \in \left(0, \mathbf{\frac{1}{2}}\right)$	$\Theta\left(\frac{1}{\left(\frac{1}{2}-\alpha\right)^2}\right)$	$\Theta\left(\frac{1}{\left(\frac{1}{2}-\alpha\right)^2}\right)$	[Goldreich Levin 89]	[Blinovsky 86, Guruswami Vadhan 10, Grinberg Shaltiel Viola 18]
Erasures	$\alpha \in (0,1)$	$O\left(\frac{1}{1-\alpha}\right)$	$\Theta\left(\frac{1}{1-\alpha}\right)$	**new**	Implicit in [Grinberg Shaltiel Viola 18]

An improvement in dependence on α was suggested by Venkat Guruswami

How does separating erasures from errors in local list decoding help with separating them in property testing?

3CNF Properties: Hard to Test, Easy to Decide

- Formula ϕ_n : 3CNF formula on n variables, $\theta(n)$ clauses
- Property $P_{\phi_n} \subseteq \{0,1\}^n$: set of satisfying assignments to ϕ_n

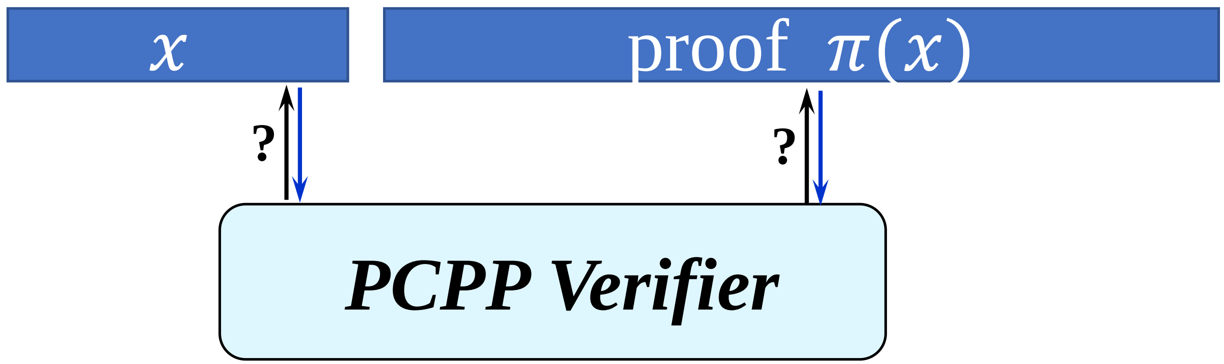
Theorem [Ben-Sasson Harsha Raskhodnikova 05]

For sufficiently small **ε**,
ε-testing P_{ϕ_n} requires $\boldsymbol{\Omega(n)}$ queries.

- P_{ϕ_n} decidable by an $\mathbf{O}(\boldsymbol{n})$-size circuit.

Testing with Advice: PCPs of Proximity (PCPPs)

[Ergun Kumar Rubinfeld 99, Ben-Sasson Goldreich Harsha Sudan Vadhan 06, Dinur Reingold 06]

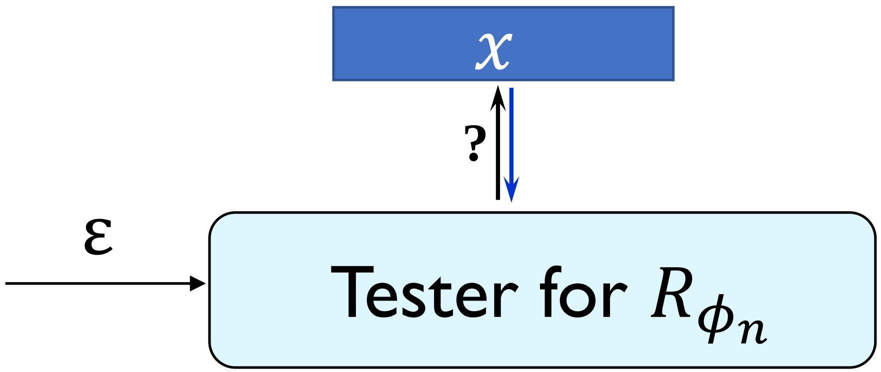


- If x has the property, then $\exists\pi(x)$ for which verifier accepts.
- If x is ε-far, then $\forall\pi(x)$ verifier rejects with probability $\geq 2/3$.

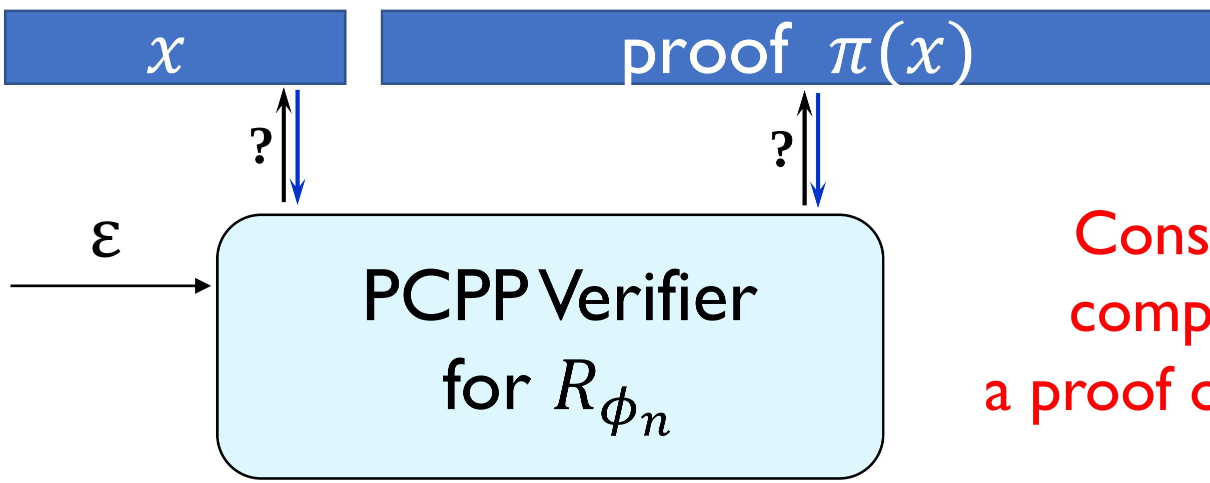
Theorem

Every property decidable with a circuit of size $\boldsymbol{m}$
has PCPP with proof length $\tilde{\boldsymbol{O}}(\boldsymbol{m})$ and constant query complexity.

Testing 3CNF Properties with/without a Proof



Need $\Omega(n)$
queries to test
without a proof



Constant query
complexity with
a proof of length $\tilde{O}(n)$

Separating Property

x^{r}	$\mathrm{Enc}(x \circ \pi(x))$

- x satisfies the hard 3CNF property
- r is the number of repetitions (to balance the lengths of 2 parts)
- $\pi(x)$ is the proof on which the PCPP verifier accepts x
- Enc uses a **locally list erasure-decodable** error-correcting code
  - E.g., Hadamard;
  - Codes with a better rate imply a stronger separation.

Separating Property: Erasure-Resilient Testing

x^{r}	Hadamard$(x \circ \pi(x)\ )$

Idea: If a constant fraction (say, 1/4) of the encoding is preserved, we can locally list erasure-decode.

Erasure-Resilient Tester

1. Locally list erasure-decode Hadamard to get a list of algorithms.
2. For each algorithm, check if:
   - the plain part is x^r by comparing u.r. bits with the corresponding bits of the decoding of x
   - PCPP verifier accepts $x \circ \pi(x)$
3. Accept if, for some algorithm on the list, both checks pass.

Constant query complexity.

Separating Property: Hardness of Tolerant Testing

x^{r}	Hadamard$(x \circ \pi(x)\ )$

Idea: Reduce standard testing of 3CNF property to tolerant testing of the separating property.

- Given a string x, we can simulate access to

x^{r}	00000 … 00000

- All-zero string is Hadamard$(x \circ \pi(x))$ with 1/2 of the encoding bits corrupted!
- Testing 3CNF property requires $\Omega(n)$ queries, where $n = |x|$.
  The input length for separating property is $N \approx 2^{cn}$.

$\Omega(n) \approx \Omega(\log N)$ *queries are needed.*

What We Proved

The separating property is

- erasure-resiliently testable with a constant number of queries,
- but requires $\widetilde{\Omega}(\log N)$ queries to tolerantly test.

Tolerant testing is harder than
erasure-resilient testing in general.

Strengthening the Separation: Challenges

If there exists a code that is locally list decodable from an $\alpha < 1$ fraction of erasures with

- list size ℓ and number of queries q that only depend on α
- inverse polynomial rate

then there is a stronger separation: constant vs. N^c.

The existence of such a code is an open question.

The corresponding question for the case of errors is the holy grail of research on local decoding.

Strengthening the Separation: Main Ideas

- **Observation:** Queries of the PCPP verifier can be made nearly uniform over proof indices
  [Dinur 07] + [Ben-Sasson Goldreich Harsha Sudan Vadhan 06, Guruswami Rudra 05]
  – No need to decode every proof bit

- **Idea:** Encode the proof with approximate LLDCs that decode a constant fraction of proof bits correctly.
  – Approximate LLDCs of inverse-polynomial rate are known [Impagliazzo Jaiswal Kabanets Wigderson 10]
  – Approximate LLDCs $\Rightarrow$ approximate locally list erasure-decodable codes of asymptotically the same rate

Open Questions and Directions

- Even stronger separation -- constant vs. linear?
- Separation between errors and erasures for a "natural" property?
- Are locally list erasure-decodable codes provably better than LLDCs?
  - We showed it for Hadamard in terms of ℓ and q.
  - Same question for the approximate case.
- Constant-query, constant list size, local list erasure-decodable codes with inverse polynomial rate?