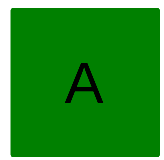
Hard instances for satisfiability and quasi-one-way functions

Andrej Bogdanov and Kunal Talwar and Andrew Wan

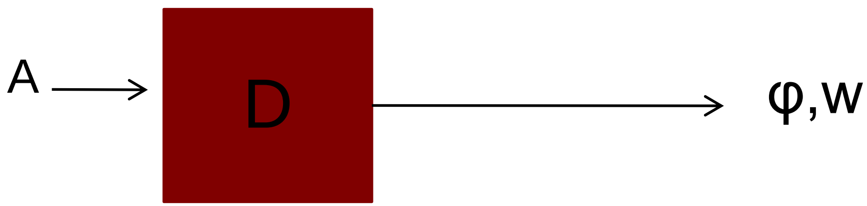
“Dreambreakers”



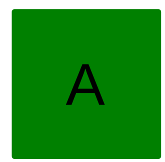
SAT algorithm

If P≠NP, then A must fail.

“Dreambreakers”

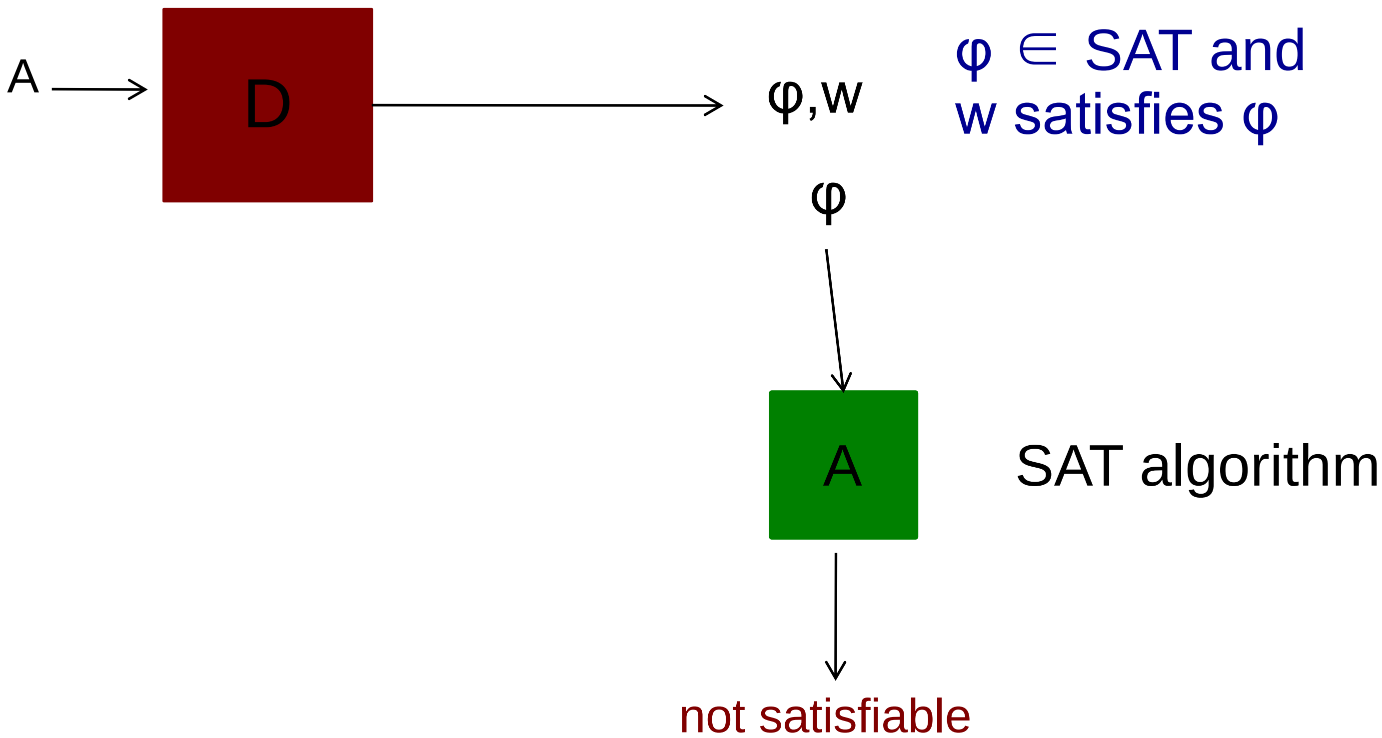


$\varphi \in$ SAT and
w satisfies φ

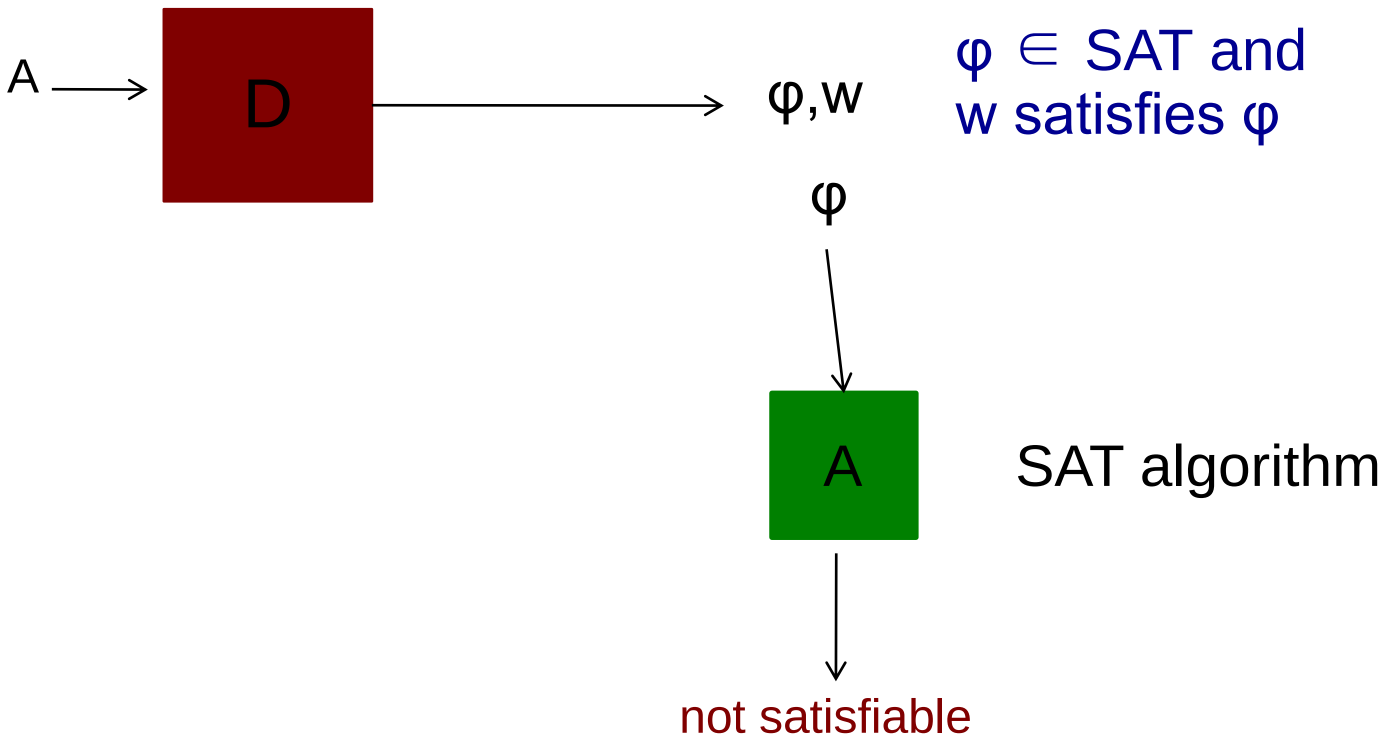


SAT algorithm

“Dreambreakers”



“Dreambreakers”



“SAT solvers” are widely used in software verification, AI, and operations research.

This work

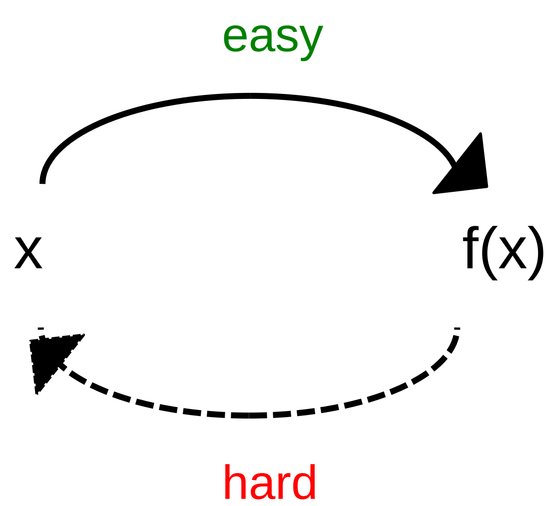
- Construct dreambreakers
- Explore relationship to cryptography

Outline

- Cryptographic motivation
- Construction of dreambreakers
- Dreambreakers and OWFs
- Dreambreakers and PRGs

Cryptography and Hardness Assumptions

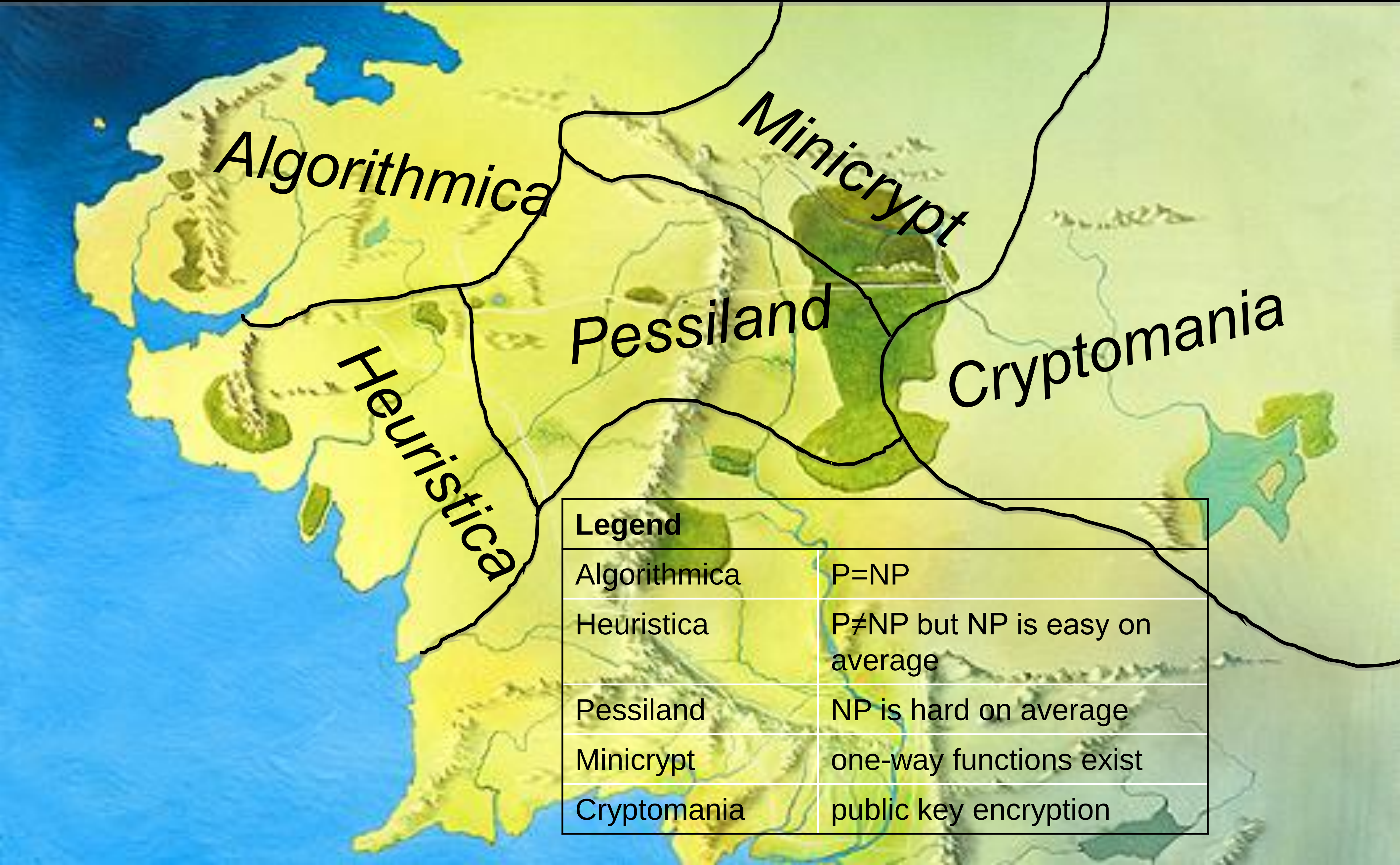
One-way function



Requires P≠NP

Does P≠NP imply cryptography?

Impagliazzo’s five worlds



Legend	
Algorithmica	P=NP
Heuristica	P≠NP but NP is easy on average
Pessiland	NP is hard on average
Minicrypt	one-way functions exist
Cryptomania	public key encryption

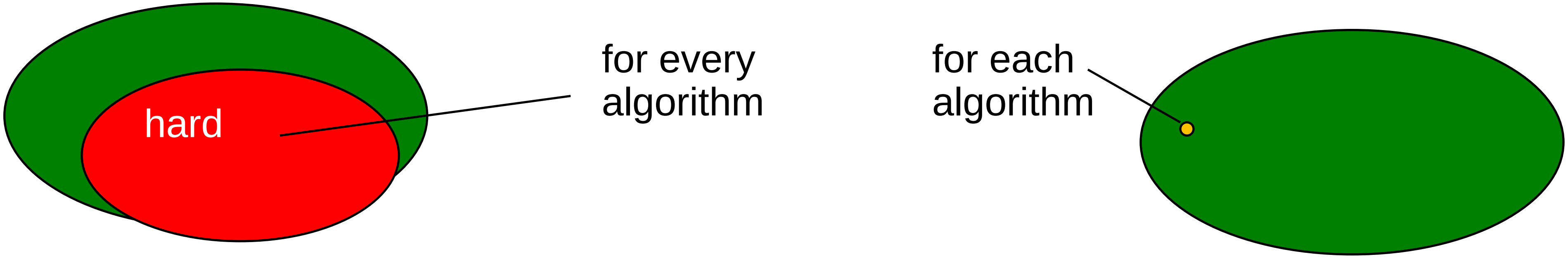
Impagliazzo's five worlds

Can we rule out Heuristica or Pessiland?

Cryptography and Hardness Assumptions

Enormous obstacle: Ruling out Heuristica [FF93,BT03,AGGM06]

i.e., obtaining *average*-case hardness from *worst*-case hardness



Other Barriers: ruling out Pessiland

[Imp95] Pessiland--average-case hardness but no cryptography.

May have a hard distribution over SAT, but how can we turn this into a one-way function?

Other Barriers: ruling out Pessiland

[Imp95] Pessiland--average-case hardness but no cryptography.

May have a hard distribution over SAT, but how can we turn this into a one-way function?

Fact: OWFs imply ability to sample hard instances of problems AND their solutions.

Given OWF f choose random “solution” x, and “problem” f(x)

Other Barriers: ruling out Pessiland

[Imp95] Pessiland--average-case hardness but no cryptography.

May have a hard distribution over SAT, but how can we turn this into a one-way function?

Fact: OWFs imply ability to sample hard instances of problems AND their solutions.

Question: If we can sample hard instances, can we sample their solutions?

P≠NP and Heuristica revisited

[GST05]

P ≠ NP, but algorithm A that solves SAT on every efficiently samplable distribution D?

P≠NP and Algorithmica revisited

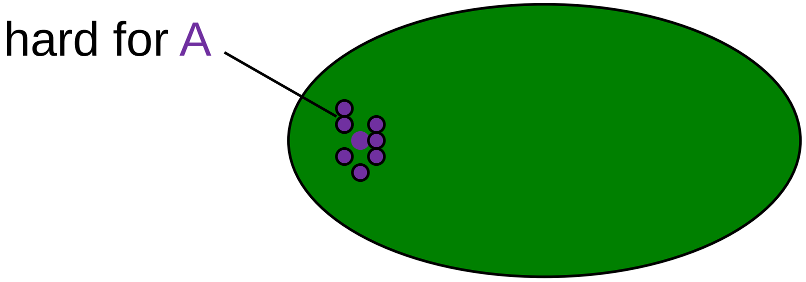
[GST05]

P ≠ NP, but algorithm A that solves SAT on every efficiently samplable distribution D?

Thm: If P ≠ NP, for any decision algorithm A, there is an efficiently samplable distribution D_A that is hard for A.

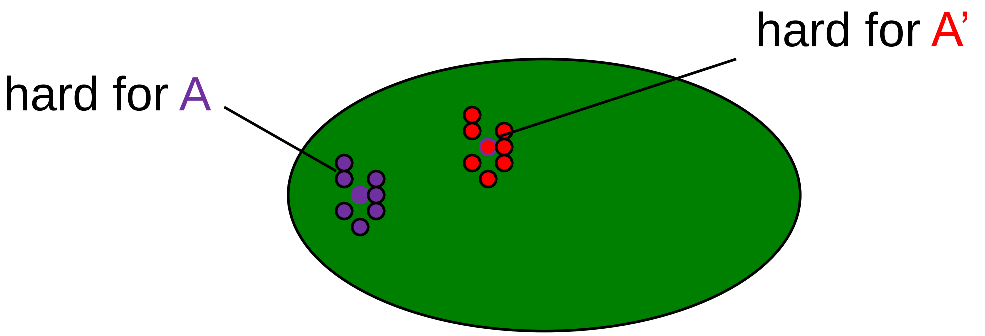
P≠NP and Algorithmica revisited

Thm: [GST05] If P ≠ NP, for any decision algorithm A, there is an efficiently samplable distribution D_A that is hard for A



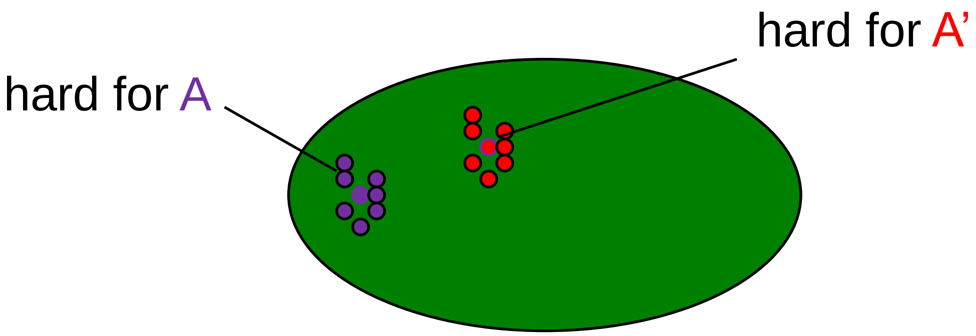
P≠NP and Algorithmica revisited

Thm: [GST05] If P ≠ NP, for any decision algorithm A, there is an efficiently samplable distribution D_A that is hard for A, for A'



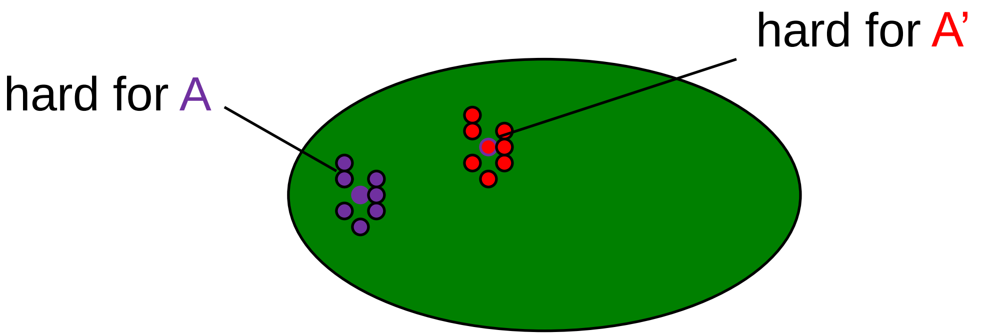
P≠NP and Algorithmica revisited

Thm: [GST05] If P ≠ NP, for any decision algorithm A, there is an efficiently samplable distribution D_A that is hard for A, for A', etc.



P≠NP and Algorithmica revisited

Thm: [GST05] If P ≠ NP, for any decision algorithm A, there is an efficiently samplable distribution D_A that is hard for A, for A', etc.



Open Question[GST05]*: (dreambreakers) Can we sample hard formulas AND their satisfying assignments?

* suggested by Adam Smith

Summary: cryptographic motivation

Does P≠NP imply OWFs?

If P≠NP: can we sample hard instances, and
can we sample their solutions?

If P≠NP: can we weakly sample hard instances [GST05], and
can we weakly sample their solutions?

Summary: cryptographic motivation

Does P≠NP imply OWFs?

If P≠NP: can we sample hard instances, and
can we sample their solutions?

If P≠NP: can we weakly sample hard instances [GST05], and
can we weakly sample their solutions?

Can we build dreambreakers?

Our work: construct dreambreakers

Thm: If P≠NP, there is poly-time procedure D, for any poly-time search algorithm A :

$$D(1^n, 1^{t(n)}, A) \rightarrow (\varphi, w) \qquad |\varphi|=n$$

And for infinitely many n,

- φ satisfied by w, and
- A(φ)=0

Our work: dreambreakers exist

Thm: If P≠NP, there is poly-time procedure D, for any poly-time search algorithm A :

$D(1^n, 1^{t(n)}, A) \rightarrow (\varphi, w)$ $|\varphi|=n$

And for infinitely many n,

- φ satisfied by w, and
- $A(\varphi)=0$

Probabilistic version

Corollary: (Quasi-hard samplers) Sampler S which takes $1^n,1^{t(n)}$ and outputs (φ,w) hard for every p.p.t. running in time t(n).

Sampling algorithms

In [GST05]: Diagonalize--Run A on formula that describes success of A on smaller instances.

Use A to find instances on which it fails.

We also use A to find solutions to instances on which it fails!

This work

- Construct dreambreakers
- Explore relationship to cryptography

Outline

- Cryptographic motivation
- Construction of Dreambreakers
- Dreambreakers and OWFs
- Dreambreakers and PRGs

Quasi-hard samplers and Cryptography

How does this relate to our cryptographic motivation?

- 'Hard' distribution, but sampler S takes more time than the adversaries it fools
- compare to sampling in fixed polynomial time to fool all poly-time algorithms
- much weaker notion of avg case hardness

Quasi-hard samplers and Cryptography

How does this relate to our cryptographic motivation?

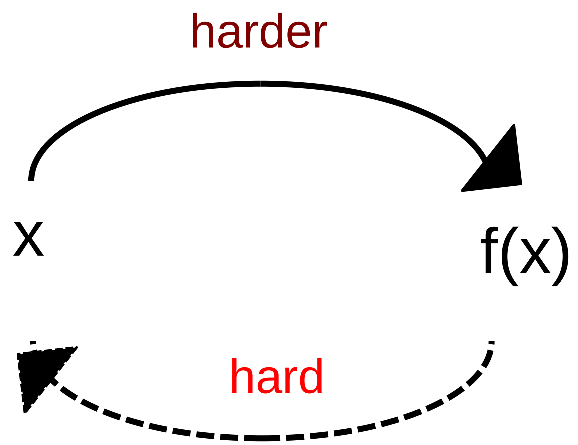
- ‘Hard’ distribution, but sampler S takes more time than the adversaries it fools
- compare to sampling in fixed polynomial time to fool all poly-time algorithms
- much weaker notion of avg case hardness

[GT07] This weaker notion still ‘contradicts’ barriers outlined in [BT,FF]

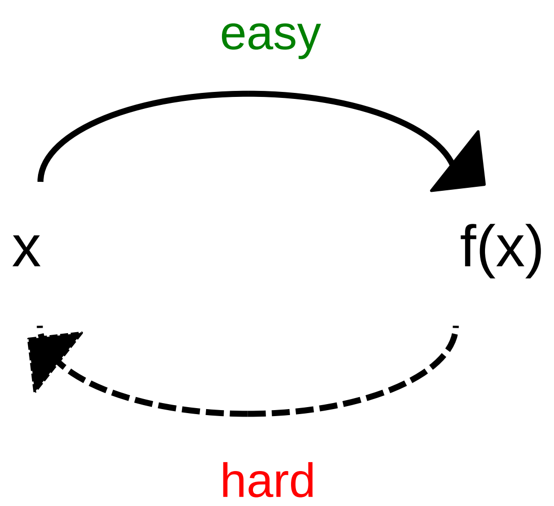
Can we achieve cryptographic primitives for this weaker notion of avg case hardness? How should we define them?

Quasi-OWFs

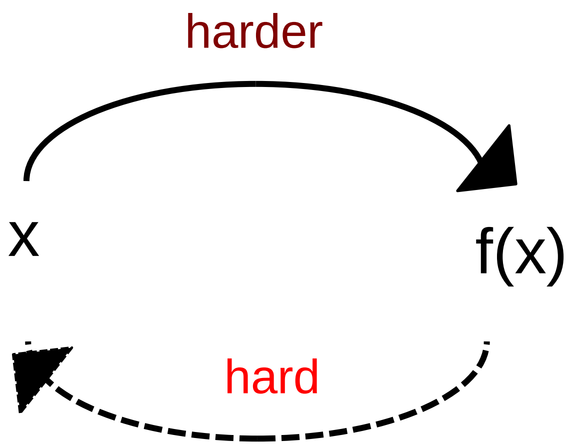
- Somewhat hard to invert
- Harder to compute



OWFs

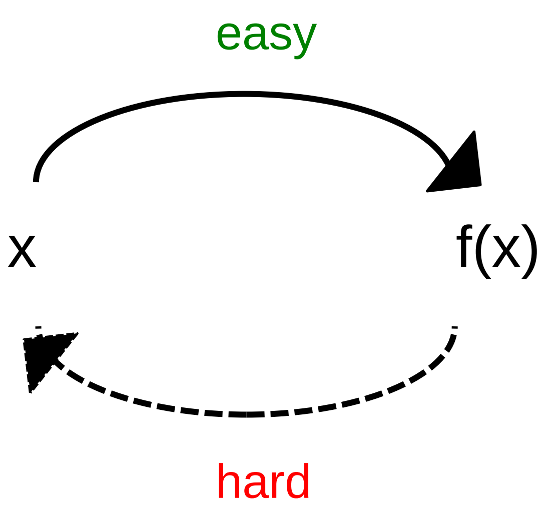


Quasi-OWFs



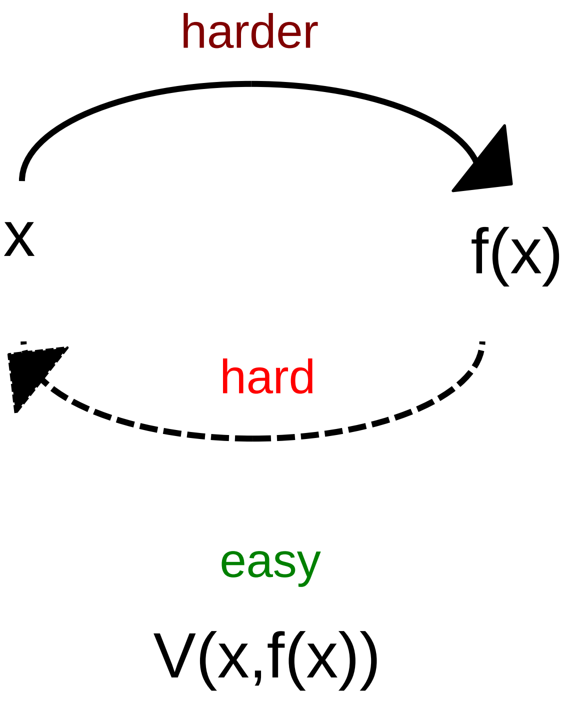
- Somewhat hard to invert
- Harder to compute
- Useless?

OWFs

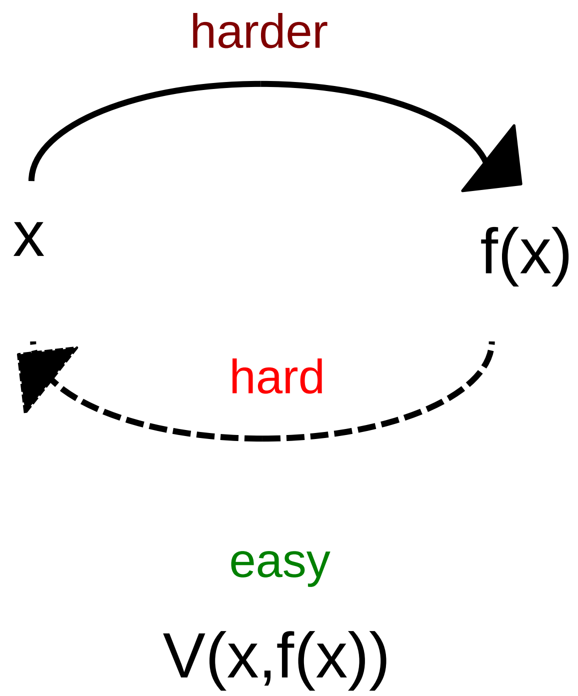


Quasi-OWFs

- Somewhat hard to invert
- Harder to compute
- Easy to verify



Quasi-OWFs



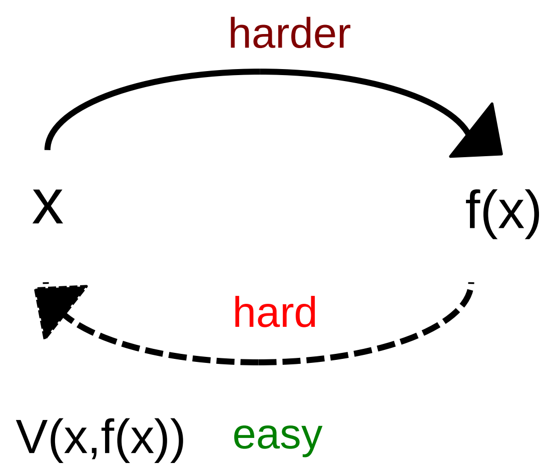
- Somewhat hard to invert
- Harder to compute
- Easy to verify

Without verifier condition, exist unconditionally

Quasi-one-way functions imply P≠NP

Non-trivial aspect of easiness-hardness contrast

Quasi-OWFs



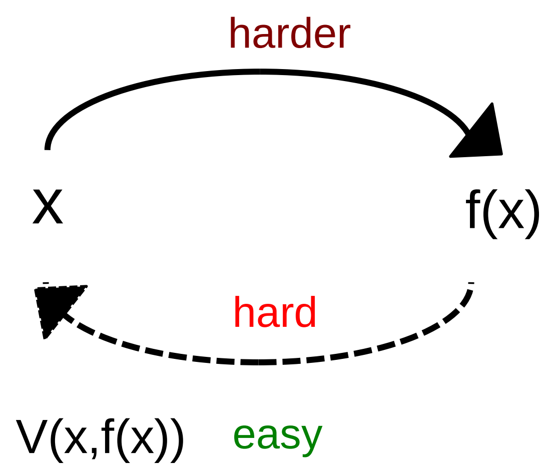
Def: Fix a polynomial $t_V(n)$ and let $t(n)>t_V(n)$. A poly-time function f is quasi-one-way against time $t(n)$ with verifier V (running in time $t_V(n)$) if for every x:

(easy to verify) $V(x,f(x))=1$,

and for every algorithm A running in time $t(n)$,

(hard to invert) $\Pr_x[V(A(f(x)),f(x))=1]<1/t(n)$.

Quasi-OWFs



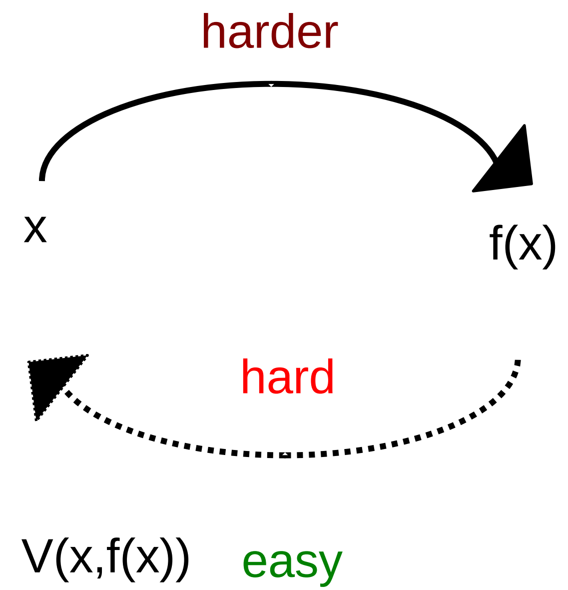
Def: Fix a polynomial $t_V(n)$ and let $t(n)>t_V(n)$. A poly-time randomized function f is quasi-one-way against time $t(n)$ with verifier V (running in time $t_V(n)$) if for every x:

(easy to verify) $V(x,f(x))=1$,

and for every probabilistic algorithm A running in time $t(n)$,

(hard to invert) $\Pr_{x,A}[V(A(f(x)),f(x))=1]<1/t(n)$.

Quasi-OWFs



Thm: If NP $\underline{\not\subset}$ BPP then for any poly t(n), quasi-OWFs against time t(n) exist.

f: $\{0,1\}^n \rightarrow \{0,1\}^{2n}$

Use quasi-hard sampler S:

$S(1^{p(t(n))}) \rightarrow \varphi, w$

$f(r) = (\varphi\ , w+r)$

Verifier: V(r,(φ,w)) accepts if (φ,w)='0' or r+w satisfies φ

This work

- Construct dreambreakers
- Explore relationship to cryptography

Outline

- Cryptographic motivation
- Construction of Dreambreakers
- Dreambreakers and OWFs
- Dreambreakers and PRGs

Quasi-OWFs and PRGs

PRG with stretch k: $G:\{0,1\}^n \rightarrow \{0,1\}^{n+k}$ $G(U_n) \approx U_{n+k}$

Generator more time than adversary

Well motivated application: algorithmic derandomization

Quasi-OWFs and PRGs

PRG with stretch k: $G:\{0,1\}^n \rightarrow \{0,1\}^{n+k}$ $G(U_n) \approx U_{n+k}$

Generator more time than adversary

Well motivated application: algorithmic derandomization

PRGs against time $t(n)$ running in time $poly(t)$
implies derandomization from P≠NP

Quasi-OWFs and PRGs

PRG with stretch k: $G:\{0,1\}^n \rightarrow \{0,1\}^{n+k}$ $G(U_n) \approx U_{n+k}$

Generator more time than adversary

Well motivated application: algorithmic derandomization

Can we use quasi-one-way functions to construct PRGs?

does this follow from [HILL] or other standard constructions?

Quasi-OWFs and PRGs

Can we use quasi-one-way functions to construct PRGs?

does this follow from [HILL] or other standard constructions?

Thm: Not using standard constructions
(black box reductions from inverting to distinguishing)

Inverter needs to evaluate the OWF.

Summary/Conclusions

- Showed that dreambreakers exist, defined and constructed quasi-one-way functions
- Some methods we take for granted in normal setting (like OWF→ PRGs) don't work in this new setting

Open Problems

- Build PRGs using quasi-hard samplers?
- Applications? bit commitments, proof systems…
- Hard core predicates, uniform output, hardness amplification, stronger definitions of quasi-OWFs that give the adversary has more power?

This work

- Construct dreambreakers
- Explore relationship to cryptography

Outline

- Cryptographic motivation
- Dreambreakers and OWFs
- Quasi-OWFs and PRGs
- Construction of dreambreakers

Sampling algorithms

In [GSTS05]:

$\Phi_n \approx$ "There is a formula w_n of size n such that $A(w_n)=0$ but $SAT(w_n)=1$."

Run $A(\Phi_n)$

If succeed, can extract formula w_n of length n and witness a_n s.t.

$A(w_n)=0$
a_n satisfies w_n

Fail

Sampling algorithms

Warm up: deterministic case

In [GSTS05]:

Φ_n ≈ "There is a formula w_n of size n such that $A(w_n)=0$ but $SAT(w_n)=1$."

Run $A(\Phi_n)$

Fail

If succeed, can extract formula w_n of length n and witness a_n

with $A(w_n)=0$
a_n satisfies w_n

Φ_n satisfiable

Φ_n is a formula of size $|\Phi_n|$ on which A fails—use as witness and recurse

Sampling algorithms

Warm up: deterministic case

In [GSTS05]:

Φ_n ≈ "There is a formula w_n of size n such that $A(w_n)=0$ but $SAT(w_n)=1$."

Run $A(\Phi_n)$

If succeed, can extract formula w_n of length n and witness a_n

with $A(w_n)=0$
a_n satisfies w_n

Fail: A=0

Φ_n satisfiable

Φ_n is a formula of size $|\Phi_n|$ on which A fails— use as witness and recurse

Φ_n not satisfiable

?

Sampling algorithms

P≠NP → Φ_n is satisfiable i.o.

why would A succeed on these?

Solution: Redefine Φ_n so that when A fails on an instance of size n: all $\Phi_{n'}$ for n'>n are in SAT until A fails again.

Φ_n = "There is a formula w_N of size N for $n^{1/k} < N \leq n$ such that $A(w_N)=0$ but $SAT(w_N)=1$."

If Φ_n is of size q(n) set k so that $q(x) < (x-1)^k$

Sampling algorithms

Solution: Redefine Φ_n so that when A fails on an instance of size n: all $\Phi_{n'}$ for n'>n are in SAT until A fails again.

Φ_n = "There is a formula w_N of size N for $n^{1/k} < N \leq n$ such that $A(w_N)=0$ but $SAT(w_N)=1$."

If Φ_n is of size q(n) set k so that $q(x) < (x-1)^k$

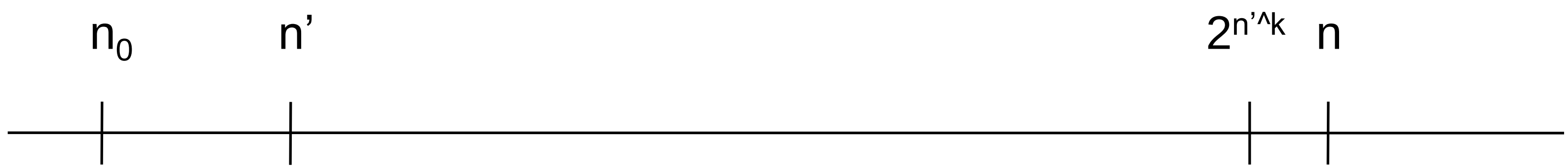
Two cases:

If A finds assignments to Φ_n infinitely often we are done.

Give a sampler for the case that A "fails" on almost all Φ_n.

Sampling algorithms

Give a sampler for the case that A fails on almost all Φ_n.



A fails on all Φ_N for $N > n_0$

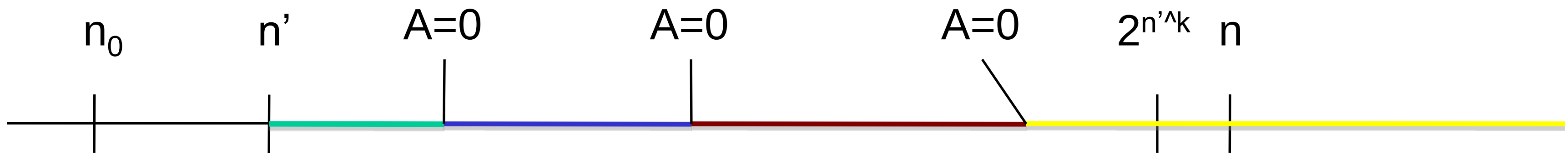
n' first input size $> n_0$ such that $\Phi_{n'} \in SAT$

Then A makes mistake on $\Phi_{n'}$, so $\Phi_{n'}$ is a good candidate witness:

If $n'' = |\Phi_{n'}|$, then $\Phi_{n'}$ is a good partial assignment for $\Phi_{n''}$

Sampling algorithms

Give a sampler for the case that A fails on almost all Φ_n.



A fails on all Φ_N for $N > n_0$

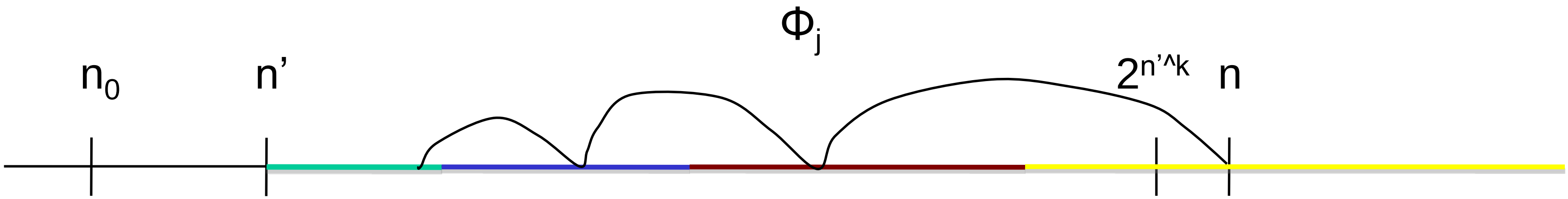
n' first input size > n_0 such that $\Phi_{n'} \in$ SAT

Sample for $n>2^{n'^k}$:

All $\Phi_{n'} \ldots \Phi_n$ are in SAT.

Sampling algorithms

Give a sampler for the case that A fails on almost all Φ_n.



A fails on all Φ_N for $N > n_0$

n' first input size $> n_0$ such that $\Phi_{n'} \in$ SAT

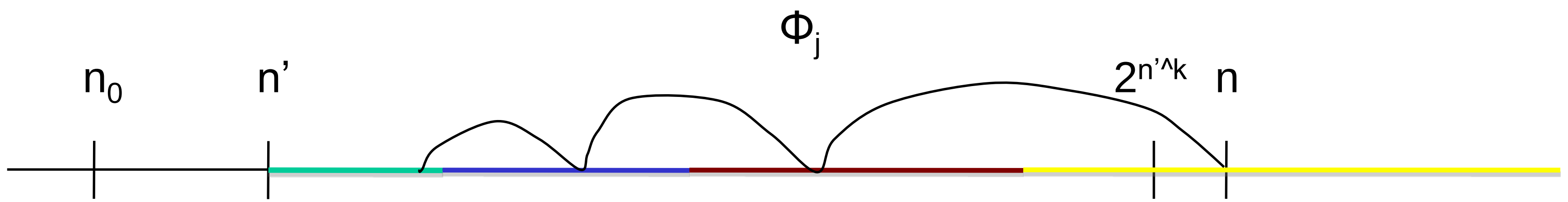
Sample for $n>2^{n'^k}$:

All $\Phi_{n'} \ldots \Phi_n$ are in SAT.

Can use Φ_j as witness for Φ_n when $n^{1/k} < q(j) \leq n$.

Sampling algorithms

Give a sampler for the case that A fails on almost all Φ_n.



A fails on all Φ_N for $N > n_0$

n' first input size $> n_0$ such that $\Phi_{n'} \in$ SAT

Sample for $n>2^{n'^k}$:

All $\Phi_{n'} \ldots \Phi_n$ are in SAT.

Can use Φ_j as witness for Φ_n when $n^{1/k} < q(j) \leq n$.

Build with smaller Φ_i until exhaustive search.

Sampling for randomized algorithms

$\Phi_{n,r} \approx$ "There is a formula w_N of size N for $n^{1/k} < N \le n$ such that $A'(w_n,r)=0$ and $SAT(w_n)=1$."

$A'(w_n,r)$ result of trying A many times. If A' fails, $Pr[A(w_n)=0]>2/3$

Two cases:

If A likely to succeed on significant fraction of $\Phi_{n,r}$ i.o. , we are done.

Build a sampler for case when A likely to fail on most $\Phi_{n,r}$ for almost all n.

Similar algorithm, choose each witness randomly.

What if A fails because $\Phi_{n,r} \notin SAT$ for most r?

Sampling if A almost always fails on most $\Phi_{n,r}$

A fails on $\Phi_{N,R}$ for N > n_0 and most R 1-p(N)

A makes mistake at length n’ → most $\Phi_{n',r'}$ in SAT. 1-r(n’)

What about $\Phi_{q(n'),r''}$-- what fraction is satisfiable?

A fails a 1-p(N) fraction of the time, but only 1-r(n’) are satisfiable