

## LECTURE 27: AUTOMATABILITY

A 1

If we want to use a proof system  $\mathcal{P}$  to solve computational problems, what properties do we want  $\mathcal{P}$  to have?

①  $\mathcal{P}$  should be POWERFUL: Given unsatisfiable formula  $F$ , want small refutations  
 $\Pi: F \vdash \perp$

②  $\mathcal{P}$  should admit EFFICIENT PROOF SEARCH: Given unsatisfiable formula  $F$ , should be possible to find refutation  $\Pi: F \vdash \perp$  quickly.

How to measure efficiency?

- If shortest proof has exponential length, then need exponential time  
So require running time  $\text{poly}(S_{\mathcal{P}}(F \vdash \perp))$
- Edge case: Consider formula

$$F = \text{GiganticMess } 1 \times 1 \dashv x$$

Constant-size proof, but we cannot find it without parsing formula

DEFINITION (AUTOMATABILITY)

Adapted from  
[BPR'00]

A proof system  $\mathcal{P}$  is AUTOMATABLE if there is an algorithm that when given unsatisfiable CNF formula  $F$  outputs a  $\mathcal{P}$ -refutation of  $F$  in time  $\text{poly}(S(F) + S_{\mathcal{P}}(F \vdash \perp))$ .

Can also study more generous notions of automatability in quasi-polynomial time, et cetera

Are the proof systems we have studied  
automatable

A II

### WIDTH-/DEGREE-AUTOMATABILITY

$n = \# \text{variables}$

If  $F$  has a resolution, Nullstellensatz, or polynomial calculus refutation in width/degree  $d$ , then such a refutation can be found in time  $n^{O(d)}$

But there are formulas with refutation width/degree  $\Omega(\sqrt{n})$  for resolution and PC and degree  $\Omega(n/\log n)$  for NS but polynomial refutation size (in fact, linear in formula size)

[Alekhnovich - Razborov '08]

Resolution is not automatable unless parameterized complexity hierarchy collapses

[Galesi - Lauria '10]

Polynomial calculus is not automatable under same assumptions

A proof system is WEAKLY AUTOMATABLE if it is polynomially simulated by an automatable proof system

For any proof system that is closed under restrictions, weak automatability implies feasible interpolation [BPR '00]

For strong enough proof systems

(that have short proofs of their own soundness)

weak automatability is equivalent to

feasible interpolation

[Pudlák '03]

(Frege and extended Frege)

Under cryptographic assumptions

- bounded-depth Frege
- Frege
- extended Frege

do not have feasible interpolation,  
and so are not weakly automatable

[KP '98, BPR '00, BDGM '04]

Breakthrough by Tocino & Müller [TM20]

Resolution is not automatable  
unless  $NP \subseteq P$

(Optimal assumptions: If  $NP \subseteq P$ , then  
resolution is automatable)

Has led to other non-automatability  
results for

- Nullstellensatz & polynomial calculus

[dRGNPRS '21]

- cutting planes [GKMP '20]

-  $k$ -DNF resolution [Garlik '20]

- tree-like resolution (under ETH)  
[de Rezende '21]

OPEN FOR: Sherali-Adams & sum-of-squares

# What we would like to cover today

A IV

**THEOREM 1** [dRGNPRS '21, building on AM '20]

There is a poly-time algorithm  $\mathcal{A}$  that

- when given 3-CNF formula  $F$  over  $n$  variables
- outputs CNF formula  $\mathcal{A}(F)$  such that  
for  $\mathcal{P}$  = resolution, polynomial calculus,  
or Nullstellensatz:
  - if  $F$  is satisfiable, then  $\mathcal{A}(F)$  has  $\mathcal{P}$ -refutation of size  $n^{O(1)}$
  - if  $F$  is unsatisfiable, then  $\mathcal{A}(F)$  requires  $\mathcal{P}$ -refutations of size at least  $\exp(n^{o(1)})$

Define

QP: problems solvable in time  $\exp(\log^{0.4} n)$   
SUBEXP:  $\exp(n^{o(1)})$

**COROLLARY 2**

For  $\mathcal{P}$  = resolution, polynomial calculus,  
or Nullstellensatz:

- (a)  $\mathcal{P}$  is not automatable in polynomial time unless  $NP \subseteq P$
- (b)  $\mathcal{P}$  is not automatable in quasi-polynomial time unless  $NP \subseteq QP$
- (c)  $\mathcal{P}$  is not automatable in subexponential time unless  $NP \subseteq \text{SUBEXP}$

# Proof sketch for corollary

A D

Suppose  $P$  is automatable.

Use proof search algorithm  $S$  to solve 3-SAT

Given 3-CNF formula  $F$

Compute CNF formula  $\delta(F)$

Run  $S$  on  $\delta(F)$  with polynomial time-out

Case analysis:

(i)  $F$  satisfiable:

Then  $\exists$  short  $P$ -refutation of  $\delta(F)$

$S$  will find this refutation

(ii)  $F$  unsatisfiable

Then there is no short  $P$ -refutation,  
so  $S$  will time out.

This decides whether  $F \in 3\text{-SAT}$  in  
polynomial time

qed

Remark

Algebraic results hold over any field  
and with or without dual variables.

Observation

When  $F$  satisfiable, short refutations of  
 $\delta(F)$  must require large width/degree  
Otherwise width-/degree-bounded  
search would find them efficiently.