ALGORITHMIC COMPLEXITY OF CERTIFIED UNSATISFIABILITY

Albert Atserias

Universitat Politècnica de Catalunya
Barcelona

I.E., COMPLEXITY OF PROOF SEARCH

Albert Atserias

Universitat Politècnica de Catalunya
Barcelona

SAT-solvers revolution (since early 2000's)

SAT-solvers "routinely" find:

satisfying assignments

or

proofs of unsatisfiability

For formulas with 1000's of variables:

search space is RIDICULOUSLY BIG!

[MS'99, Chaff 2001, ...]

“200 TB maths proof is largest ever”

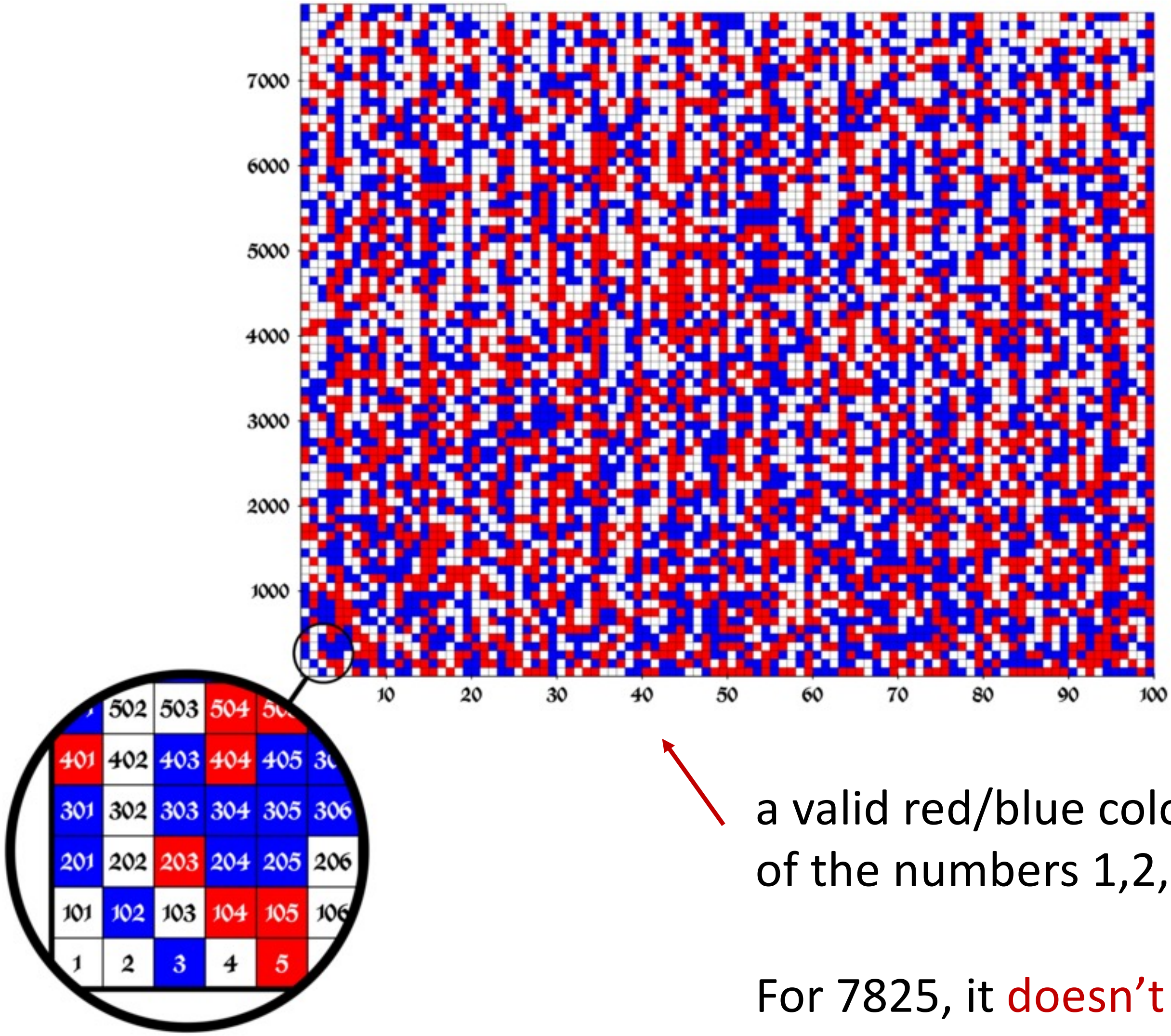
[Nature 2016]

Theorem [Heule, Kullmann, Marek 2016]

The numbers 1,...,7825 cannot be partitioned into two parts each without Pythagorean triples.

But the numbers 1,...,7824, can.

$a^2 + b^2 = c^2$



$a^2 + b^2 \neq c^2$

$a^2 + b^2 \neq c^2$

a valid red/blue coloring (white = either) of the numbers 1,2,...,7824.

For 7825, it doesn't exist.

[Source: Wikipedia]

AUTOMATABILITY

Definition of automatability

Def: P is AUTOMATABLE in polynomial time

if

an algorithm finds P-proofs in time polynomial in the size of smallest P-proof

[Bonet, Pitassi, Raz 97]

SAT-solver vs PROOF-searcher

evaluated through benchmarking

evaluated through provable guarantees

Moshe Vardi "For the SAT revolution to continue unabated, we must focus also on understanding, not only on benchmarking."

[Vardi, CACM 2014]
restated by [Sakallah, Simons 2023]

Tree-like Resolution

Theorem [Beame, Pitassi 98]

Tree-like Resolution is automatable in time $n^{O(\log s)}$

number of variables

size of smallest tree-like refutation

General Resolution

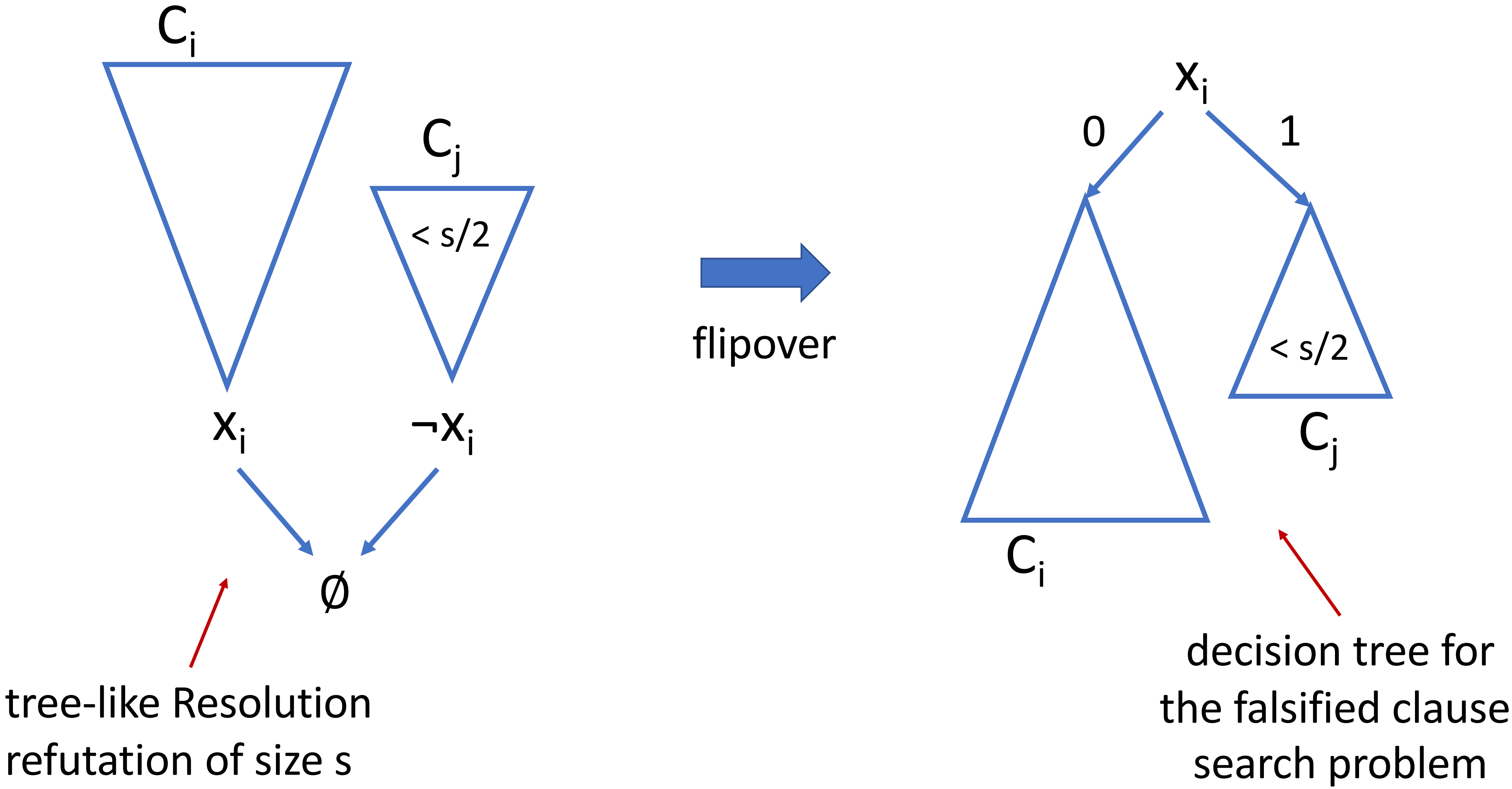
Theorem [Ben-Sasson, Wigderson 99]

Resolution is automatable in time $n^{O(\sqrt{n \log s}+k)}$

for $s = \mathrm{poly}(n), k = 3$
this is $\exp(n^{1/2} \log(n)^{3/2})$.
Compare with ETH.

width of
initial clauses

Beame-Pitassi Algorithm



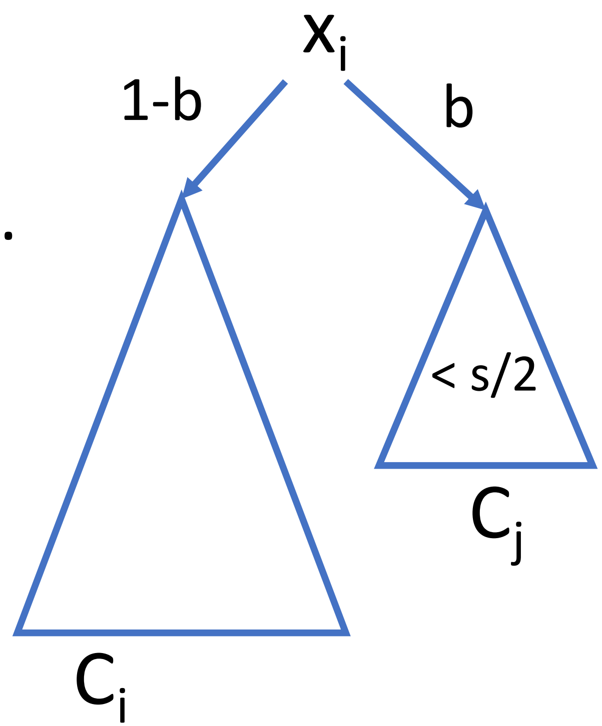
Algorithm

Given F and s.

Guess i and b and recurse on $F[x_i=b]$ and $s/2$.

Then recurse on $F[x_i=1\text{-}b]$ and $s\text{-}1$.

Subtle: Don't know
if the guess that worked
is the root of the optimal tree!



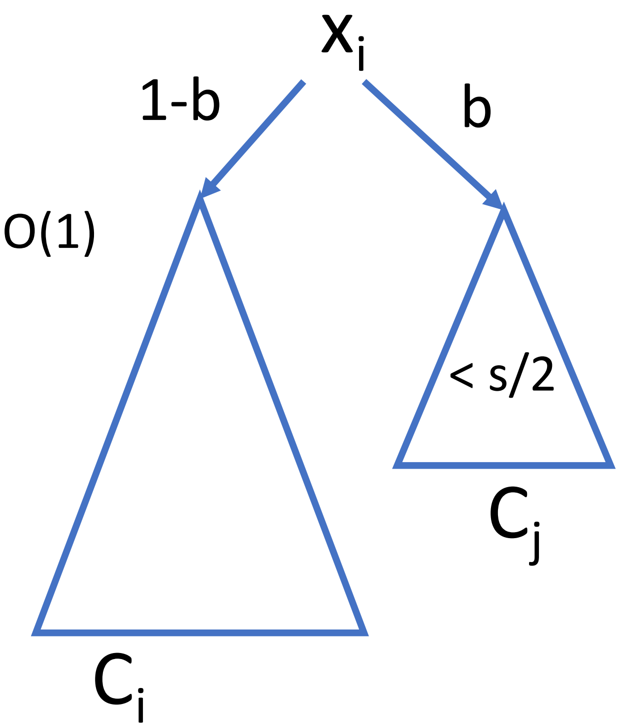
Analysis

number of variables

target size

$$R(n, s) = 2n\, R(n-1, s/2) + R(n-1, s-1) + n^{O(1)}$$

number of choices in guess



Solution: $n^{O(\log s)}$

Proof Searchers

Restatment: There is a proof-searcher for tree-like Resolution with quasipolynomial-time $n^{O(\log s)}$ guarantee.

Restatement: There is a proof-searcher for Resolution with subexponential-time $n^{O(\sqrt{n \log s})}$ guarantee.

Indeed, CDCL (with enough restarts, enough random decisions, and full memory) achieves this!

[AFT'2011]

FEASIBLE INTERPOLATION

Craig Interpolants

$$F(x, y) \wedge G(x, z)$$

suppose this is unsatisfiable.

$$\neg INT(x) \rightarrow \neg F(x, y)$$

$$INT(x) \rightarrow \neg G(x, z)$$

Then these are tautologies.

INT(x) tells which one is unsatisfiable, for each given x.

Interpolants in graph theory

$\mathrm{CLIQUE}_{k+1}(x, y)$:= "y codes a $k+1$-clique of x"
$\mathrm{COL}_k(x, z)$:= "z codes a proper k-coloring of x"

x codes a graph

$$\mathrm{CLIQUE}_{k+1}(x, y) \wedge \mathrm{COL}_k(x, z)$$

unsatisfiable
(by the PHP)

What are its interpolants?

“y is k+1-clique of x” $\wedge$ “z is k-coloring of x”

$\neg \mathrm{INT}_k(x) \rightarrow$ “$\omega(x) \leq k$”

$\mathrm{INT}_k(x) \rightarrow$ “$\chi(x) > k$”

E.g. Lovász’s Theta “$\vartheta(x) > k$”

Interpolants in Cryptography

$$\mathrm{ONE}_i(x, y) := \text{“}f(y) = x \text{ and } y_i = 1\text{”}$$

$$\mathrm{ZERO}_i(x, z) := \text{“}f(z) = x \text{ and } z_i = 0\text{”}$$

a permutation that is
easy to compute
hard to invert

unsatisfiable
since f is 1-to-1

$$\mathrm{ONE}_i(x, y) \wedge \mathrm{ZERO}_i(x, z)$$

What are its interpolants?

$$\text{“}f(y) = x \text{ and } y_i = 1\text{”} \;\wedge\; \text{“}f(z) = x \text{ and } z_i = 0\text{”}$$

$$\neg \mathrm{INT}_i(x) \;\rightarrow\; \text{“}f^{-1}(x)_i = 0\text{”}$$

$$\mathrm{INT}_i(x) \;\rightarrow\; \text{“}f^{-1}(x)_i = 1\text{”}$$

any interpolant inverts
the function (its i-th bit)

Feasible Interpolation

Def: P has feasible interpolation:

all unsatisfiable $F(x,y) \wedge G(x,z)$ have interpolants of circuit-size polynomial in the size of their smallest P-refutations.

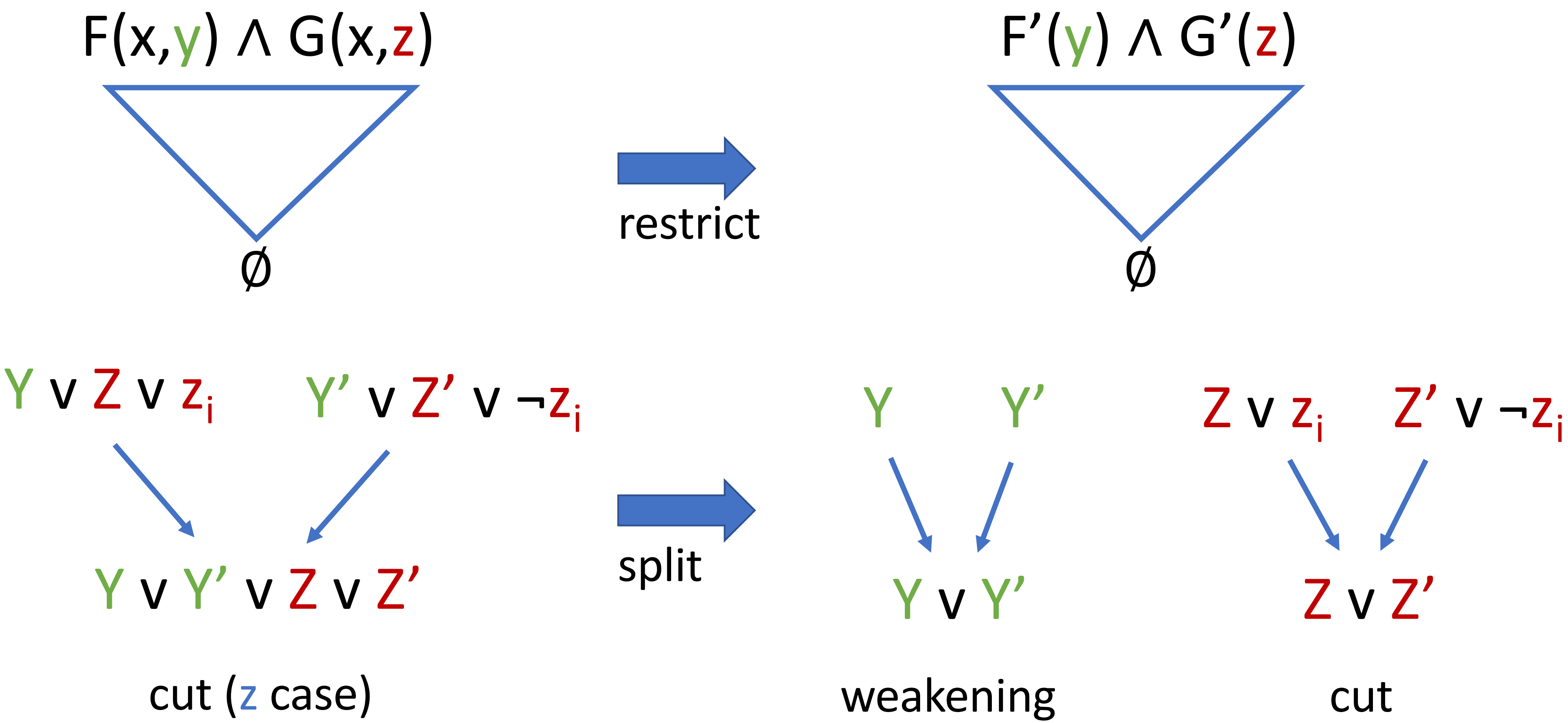
[Krajicek 1997]

Resolution has feasible interpolation

Theorem: [Krajicek 1997]
Resolution has (monotone) feasible interpolation.

Implies lower bound on CLIQUE & COL formulas
by monotone circuits lower bounds
[Razborov 1986], [Alon, Bopana 1987]

Interpolation algorithm: restrict & split



INTERPOLATION AND AUTOMATABILITY

Automatability implies Interpolation

Lemma: [Bonet, Pitassi, Raz 97]
If a proof system is automatable,
then it has feasible interpolation.

The BPR argument

suppose this
has P-refutation
of size s

$$F(x, y) \wedge G(x, z)$$

$$\mathrm{INT}(x_0) := \mathrm{REF}_{P,p(s)}(\langle G(x_0, z)\rangle, A(\langle G(x_0, z)\rangle))$$

verifier of
proof system P

If A is an automating algorithm for P
then this is an interpolant

Strong systems lack feasible interpolation

Theorem [Krajicek, Pudlak 98]

Extended Frege does not have feasible interpolation unless RSA is broken by poly-size circuits

The Krajicek-Pudlak Argument

The statements

"$RSA_i(y,k)=x$ and $y_i = 1$" $\wedge$ "$RSA_i(z,k)=x$ and $z_i = 0$"

have poly-size Extended Frege refutations.

Q.E.D.

First Non-Automatability Result: EFrege

Corollary

Extended Frege is not automatable

unless RSA is invertible in poly-time

Later improved to
Frege, TC^0-Frege, AC^0-Frege
[Bonet et al. 97, 99]

SOUNDNESS PROOFS AND AUTOMATABILITY

Interpolants of soundness statements

$\mathrm{SAT}(x, y)$:= "y codes a satisfying assignment of x"

$\mathrm{REF}_{P,s}(x, z)$:= "z codes a P-refutation of x"

codes a CNF

proof system

the size $s = |z|$

a contradiction since P is sound

$$\mathrm{SAT}(x, y) \wedge \mathrm{REF}_{P,s}(x, z)$$

Interpolants of soundness statements

$$\mathrm{SAT}(x, y) \wedge \mathrm{REF}_{P,s}(x, z)$$

$$\neg\mathrm{INT}(x) \rightarrow \neg\mathrm{SAT}(x, y)$$
$$\mathrm{INT}(x) \rightarrow \neg\mathrm{REF}_{P,s}(x, z)$$

interpolant
exists by
the soundness
of P

Sort of dual to
what a SAT-solver does!

$$\mathrm{SAT}(x, y) \wedge \mathrm{REF}_{P,s}(x, z)$$

If P is automatable
then there is a poly-time interpolant

$$\mathrm{INT}(x) := \neg\mathrm{REF}_{P,p(s)}(x, A(x))$$

polynomial runtime
of automating algorithm

automating
algorithm of P

$\mathrm{SAT}(x, y) \wedge \mathrm{REF}_{P,s}(x, z)$

If Q p-simulates P
and Q is automatable
then there is a poly-time interpolant

$$\mathrm{INT}(x) := \neg\mathrm{REF}_{Q,p(q(s))}(x, A(x))$$

polynomial loss
in automating algorithm

polynomial loss
in p-simulation

automating
algorithm of Q

[Pudlák 2001]

Weak Automatability

Theorem [Pudlák 2001]:
The following are equivalent:

(1) SAT & REF formulas for P have polytime interpolants
(2) there exists an automatable Q that p-simulates P

I.e., P is weakly automatable in Q
[A., Bonet 2003]

Resolution proofs of own soundness?

Theorem [A., Bonet 2003]
Resolution proofs of its own soundness
must be of superpolynomial in size
but poly-size Res(2)-proofs do exist!

Lower bound by reduction from
CLIQUE & COL formulas

Resolution with 2-DNFs
instead of clauses

AUTOMATING RESOLUTION IS HARD

The Alekhnovich-Razborov Theorem

Theorem [Alekhnovich-Razborov 2001]

Resolution is not automatable

unless W[P] is tractable

- relies on a strong assumption.
- best lower bound: time $n^{\log\log(n)^{0.14}}$, under ETH [Mertz-Pitassi-Wei 19]
- applies to tree-like Resolution!

Automating Resolution is NP-hard

Theorem [A., Müller 2019]

Resolution is not automatable

in polynomial-time unless P = NP

nor in subexponential-time unless ETH fails

- optimal assumption
- new method
- based on soundness proofs!

A glimpse at the proof

Find a map that takes CNFs into CNFs

$$F \xrightarrow{\text{polytime}} G$$

SMALL

F is sat $\Longrightarrow$ $\text{min-ref-size}(G) \leq |G|^{1+\varepsilon}$

F is unsat $\Longrightarrow$ $\text{min-ref-size}(G) \not\leq \exp(|G|^{\frac{1}{2}-\varepsilon})$

minimum Resolution
refutation size

BIG

The easy/hard formula

G := RREF(<F>, z)

a minor variant of REF

for poly length z

Upper bound : Uses the small soundness proof of Resolution in Res(2)!

Lower bound : Adversary argument to mimic the exponentially big refutation.

Beyond Resolution?

Thm: [Goos-Koroth-Metz-Pitassi'20]
Resolution is not weakly automatable in
Cutting Planes unless P = NP

Thm: [de Rezende-Goos-Nordström-Pitassi-Robere-Sokolov'21]
Resolution is not weakly automatable in
Nullstelensatz or Polynomial Calculus unless P = NP

Below Resolution?

Thm: [de Rezende'21]
Tree-like Resolution is not automatable
in less than quasipolynomial time
unless ETH fails

F is sat $\Longrightarrow$ min-tree-size$(G) \leq 2^{c\sqrt{N}}$
F is unsat $\Longrightarrow$ min-tree-size$(G) \not\leq 2^{dN}$

THE BIG REMAING PROBLEM

Is Resolution Weakly Automatable?

Difficulty:

Equivalent to distinguishing:

satisfiable formulas (SAT)

from

shortly refutable formulas ($\mathrm{REF}_{\mathrm{poly}}$)

both
are
problems
in NP

THE END