COMPLEXITY OF PROOF SEARCH

Albert Atserias

Universitat Politècnica de Catalunya
Barcelona

PROOF COMPLEXITY

Proof Search/Proof Size Estimation

for a language
with a semantics

given a tautology,
1) find a proof.
2) estimate the complexity of its simplest proof

for a complexity measure
for proofs

for a proof system
that is sound and complete

The Complexity of Theorem-Proving Procedures

Stephen A. Cook

University of Toronto

Summary

It is shown that any recognition problem solved by a polynomial time-bounded nondeterministic Turing machine can be "reduced" to the problem of determining whether a given propositional formula is a tautology. Here "reduced" means, roughly speaking, that the first problem can be solved deterministically in polynomial time provided an oracle is available for solving the second. From this notion of reducible, polynomial degrees of difficulty are defined, and it is shown that the problem of determining tautologyhood has the same polynomial degree as the problem of determining whether the first of two given graphs is isomorphic to a subgraph of the second. Other examples are discussed. A method of measuring the complexity of proof procedures for the predicate calculus is introduced and discussed.

certain recursive set of strings on this alphabet, and we are interested in the problem of finding a good lower bound on its possible recognition times. We provide no such lower bound here, but theorem 1 will give evidence that {tautologies} is a difficult set to recognize, since many apparently difficult problems can be reduced to determining tautologyhood. By reduced we mean, roughly speaking, that if tautologyhood could be decided instantly (by an "oracle") then these problems could be decided in polynomial time. In order to make this notion precise, we introduce query machines, which are like Turing machines with oracles in [1].

A query machine is a multitape Turing machine with a distinguished tape called the query tape, and three distinguished states called the query state, yes state, and no state, respectively. If M is a

[Cook, 1971]: ACM STOC'71

The field of mechanical theorem proving badly needs a basis for comparing and evaluating the dozens of procedures which appear in the literature. Performance of a procedure on examples by computer is a good criterion, but not sufficient (unless the procedure proves useful in some practical way). A theoretical complexity criterion is needed which will bring out fundamental limita-

[Cook, 1971]: ACM STOC'71

Comparison by p-simulation

[Reckhow 1975], [Cook-Reckhow 1974]

It's the kick off
of proof complexity

Q and P proof systems for TAUT/UNSAT

Def:

Q p-simulates P

iff

there is a poly-time computable f such that

if y is a P-proof of x, then f(y) is a Q-proof of x.

Resolution Inference Rule

given $C \lor x$ and $D \lor \neg x$ infer $C \lor D$

left premise

right premise

resolvent

Tree/Dag Proofs, Size, and Width

empty clause

$$\overbrace{C_1, \dots, C_i, \dots, C_m}^{F}: \qquad \overbrace{D_1, \dots, D_j, \dots, D_k, \dots, D_l, \dots, D_s = \emptyset}^{P}$$

Hypothesis

Left

Right

the proof-graph

$S_{Dag}(F)$:= min { length(P) : P is a Resolution refutation of F }

$S_{Tree}(F)$:= min { length(P) : P is a Tree-like Resolution refutation of F }

$W(F)$:= min { max { $|D_j|$ } : P is a (Tree-like) Resolution refutation of F }

AUTOMATABILITY

Definition of automatability

Def: P is AUTOMATABLE in time t(s)
if
an algorithm finds P-proofs in time t(s)
the size s of smallest P-proof

[Bonet, Pitassi, Raz 97]

Proof size estimation problem

Fact:
If P is automatable in time $t(s)$,
then optimal proof-size for P is $t(s)$-approximable
(in time $t(s)$).

$$opt \le estimate \le t(opt)$$

CHARACTERIZATIONS OF PROOF MEASURES

Understanding Provability

Which F have (low complexity) refutations?

Which F do <u>not</u> have (low complexity) refutations?

Answer: Those that are "locally" satisfiable

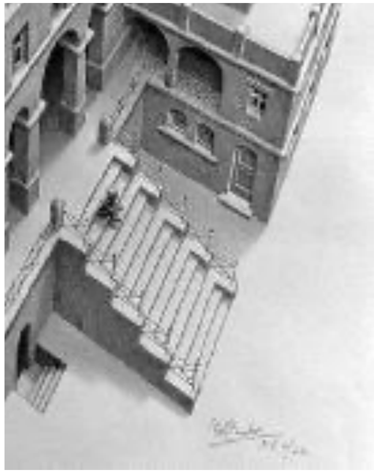
Locally Consistent Assignments

Credit:
Ascending and Descending
by M. C. Escher, 1960

Locally Consistent Assignments

Locally Consistent Assignments

Locally Consistent Assignments



Locally Consistent Assignments

Locally Consistent Assignments

Locally Consistent Assignments

The width-k Prover-Adversary game

Let F be a k-CNF with variables $[n]$, let $w \geq k+1$ an integer.

Let $M_w(n)$ = { partial truth assignments f with $|f| \leq w$ }

Prover and Adversary play a game:

- the positions are the f in $M_w(n)$,
- query moves:

  Prover queries a variable,

  Adversary assigns a value,
- shrinking moves:

  Prover suggests to discard some earlier assignments,

  Adversary happily accepts.

Goal of Prover: reach an f that falsifies a clause of F.

The width-k Prover-Adversary game

H is the
winning region
for Adversary

Def: [Kolaitis-Vardi 2000], [A.-Dalmau 2004]
A winning strategy for Adversary in the width-w game on F
is a set $H \subseteq M_w(n)$ of partial truth assignments s.t.:

1) H is non-empty,
2) f in $H \Longrightarrow f$ is consistent with F,
3) f in H and $g \subseteq f \Longrightarrow g$ in H,
4) f in H, $|\mathrm{Dom}\ f| < w$, x in $F \Longrightarrow \exists\ b$ in $\{0,1\}$ s.t. $f \cup \{ x := b \}$ in H.

Characterizes Resolution Width

Theorem: [A.–Dalmau 2004]
Let F be a k-CNF. Let $w \geq k+1$.
The following statements are equivalent:

1) there is no width-w Resolution refutation of F,
2) there is a winning strategy for Adversary in the width-w game on F.

Not a difficult theorem:
just extremely useful...

IS THERE A GOOD CHARACTERIZATION OF RESOLUTION SIZE?

The Size-Width Relationships

Theorem: [Ben-Sasson-Wigderson 2001]

Let F be a 3-CNF with n variables. Then:

$$2^{W(F)} \leq S_{\mathrm{Tree}}(F) \leq n^{W(F)}$$

$$2^{cW(F)^2/n} \leq S_{\mathrm{Dag}}(F) \leq n^{W(F)}$$

Solving for W(F) ...

Corollary: Tree-like Resolution size is $n^{O(\log s)}$-approximable.

Corollary: General Resolution size is $n^{O(\sqrt{n \log s})}$ -approximable.

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

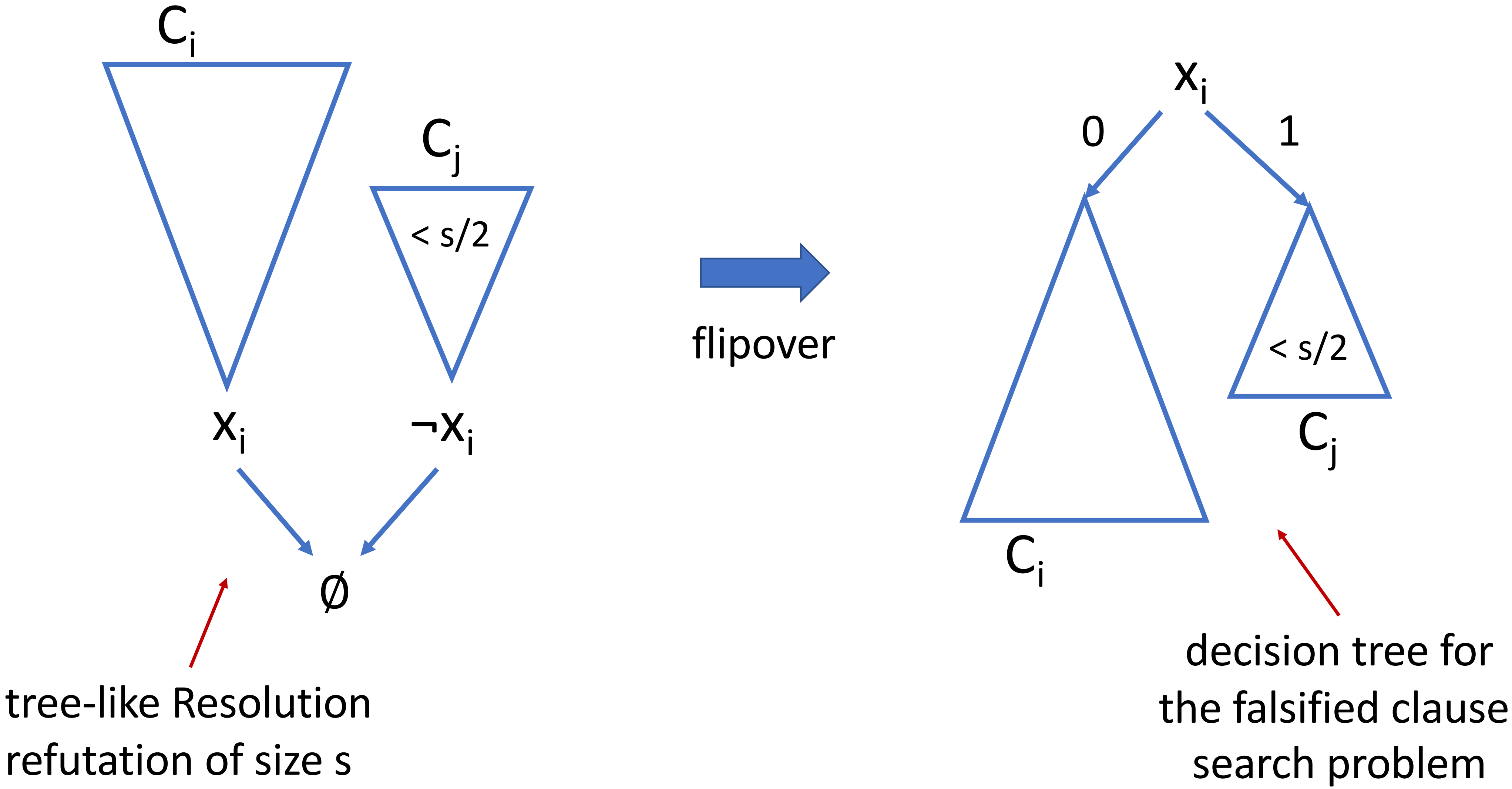
General Resolution

Theorem [Ben-Sasson, Wigderson 99]

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

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

Beame-Pitassi Algorithm



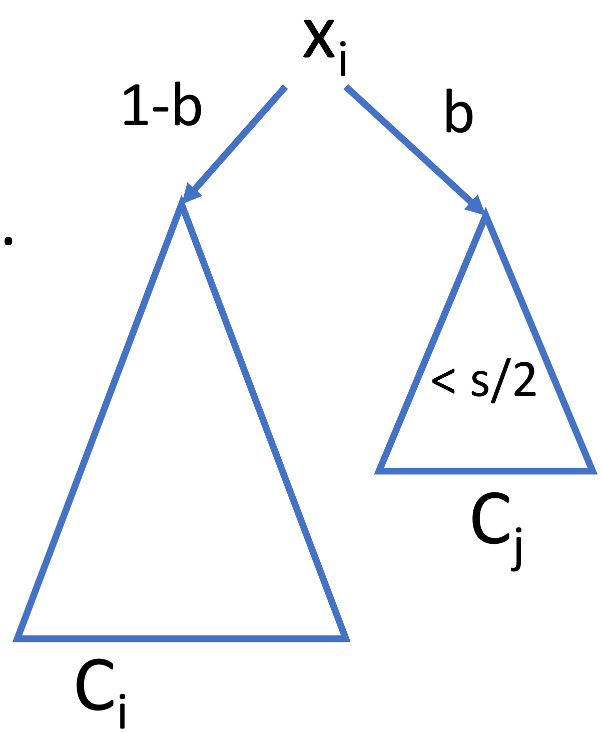
Algorithm

Given F and s.

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

Then recurse on $F[x_i=1-b]$ and $s-1$.

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



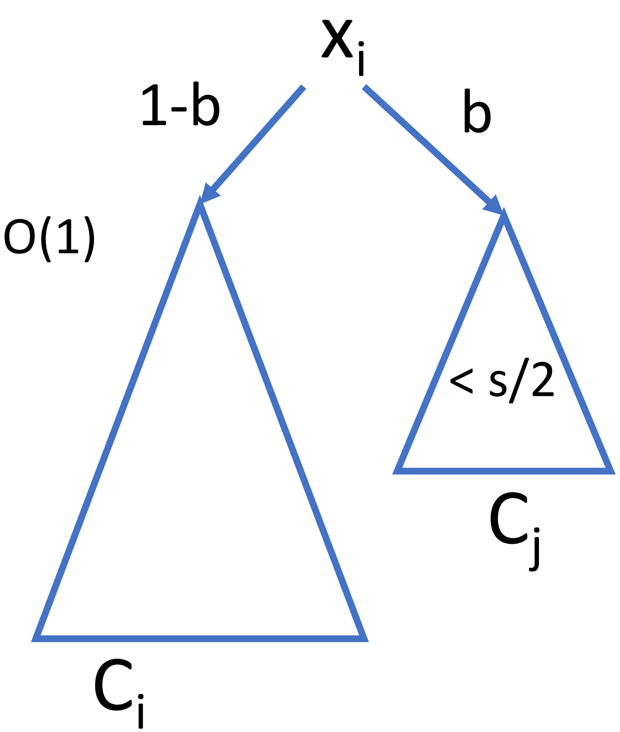
Analysis

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

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" ∧ "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?

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

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

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

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

Feasible Interpolation

Def: P has feasible interpolation:

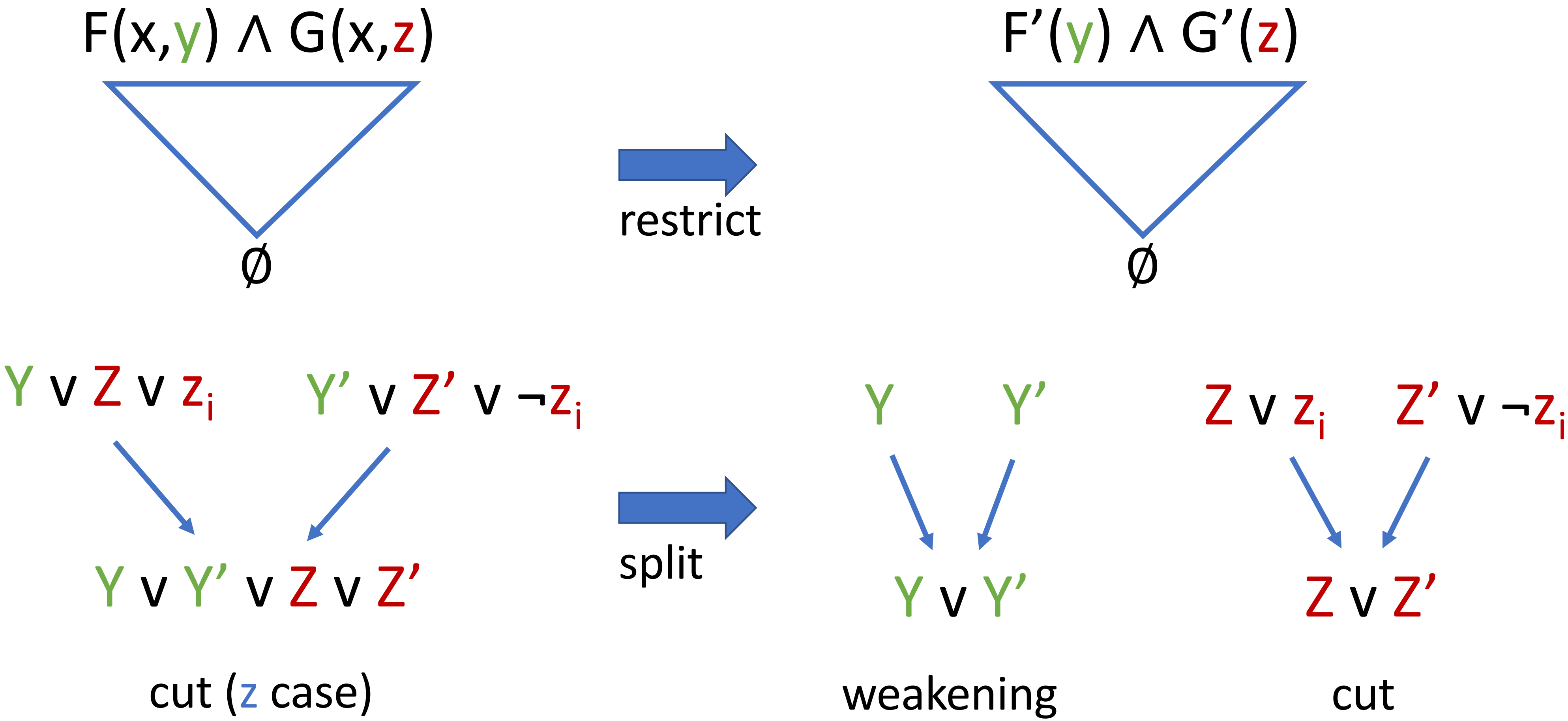
all unsatisfiable $F(x,y) \wedge G(x,z)$ have interpolants of circuit-size polynomial in the size of their smallest P-refutations.

[Krajicek 1997]

Resolution has feasible interpolation

Theorem: [Krajicek 1997]
Resolution has feasible interpolation.

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

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

$$INT(x_0) := REF_{P,p(s)}(<G(x_0, z)>, A(<G(x_0, z)>))$$

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

Well beyond resolution

Cryptographic assumption:
I.e., hardness of NP ∩ co-NP ...

SOUNDNESS PROOFS

AND

AUTOMATABILITY

Interpolants of soundness statements

SAT(x, y) := "y codes a satisfying assignment of x"

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

proof system

the size
s = |z|

codes a CNF

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

- still relied 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

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

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

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.

Below/Beyond 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$(G) \nleq 2^{dN}$

THE BIG REMAING PROBLEM

Is Resolution Weakly Automatable?

Difficulty:

Equivalent to distinguishing:

satisfiable formulas (SAT)

from

shortly refutable formulas (REF_{poly})

both are problems in NP

THE END