Effectively Polynomial Simulations

Toni Pitassi (U of Toronto)
Rahul Santhanam (U of Edinburgh)

Proof systems

Propositional

Poly time onto fn: $\{0,1\}^* \longrightarrow$ TAUT

(proofs) (prop. tautologies)

Quantified

Poly time onto fn: $\{0,1\}^* \longrightarrow$ QTAUT

(proofs) (valid quantified formulae)

Proof Systems and Complexity

- Theorem [Cook-Reckhow]: NP = coNP iff there is a propositional proof system which is *polynomially bounded* (every tautology has a proof of length polynomial in size of tautology)
- PSPACE = NP iff there is a quantified proof system which is polynomially bounded

p-Simulations

- P p-simulates Q if for all tautologies ϕ
  - If ϕ has Q-proofs of size n, then ϕ has P-proofs of size $poly(n)$
- If P p-simulates Q then proof size lower bounds for P translate to lower bounds for Q
- Extended Frege p-simulates Frege p-simulates Bounded-depth Frege p-simulates Resolution

Effectively p-simulations: Basic Idea

- Relaxed notion of simulation
- P effectively p-simulates Q if
  - ϕ has small proofs in Q $\rightarrow$ $f(\phi)$ has small proofs in P, where f is poly-time

Effectively p-simulations: Definition

Effectively p-simulation of Q by P

$<\phi,1^m> \longrightarrow$ Poly time computable f $\longrightarrow f(<\phi,1^m>)$

- For all ϕ and m, ϕ is a tautology iff $f(<\phi,1^m>)$ is a tautology
- If ϕ has Q-proofs of size at most m, then $f(<\phi,1^m>)$ has P-proofs of size at most $poly(m)$

Effectively p-simulation: Motivation

- When proof systems are used in SAT solvers, natural to allow poly-time preprocessing
- Allows us to
  - Compare proof systems of different kinds, eg. propositional vs quantified
  - Relate several pairs of proof systems not known to be related before
- Useful in studying *automatizability* (efficient proof search)

Automatizability

$\langle \phi, 1^m \rangle$ → Poly time computable f → "Proof" of ϕ, if ϕ has P-proofs of size <= m

- "Proof" might not be in P, but in a different proof system. If proof produced is a P-proof, then "strongly automatizable"

Automatizability and Complexity

- Theorem: The following are equivalent
  - Every propositional proof system is automatizable
  - Every quantified proof system is automatizable
  - P = NP

Automatizability and Effectively Polynomial Simulation

- Proposition: If P is automatizable and P effectively p-simulates Q, then Q is automatizable
- Proof: Given $<\phi,1^m>$, automatization procedure for Q runs automatization procedure for P on $f(<\phi,1^m>)$ and returns the result

Proof Systems: Hilbert-style (Propositional)

- Axioms, rules of deduction, lines of proof are propositional
- Different proof systems depending on what the lines are
  - Clauses: Resolution
  - k-DNFs: k-Res
  - AC^0: Bounded-depth Frege
  - Formulae: Frege
  - Circuits: EF

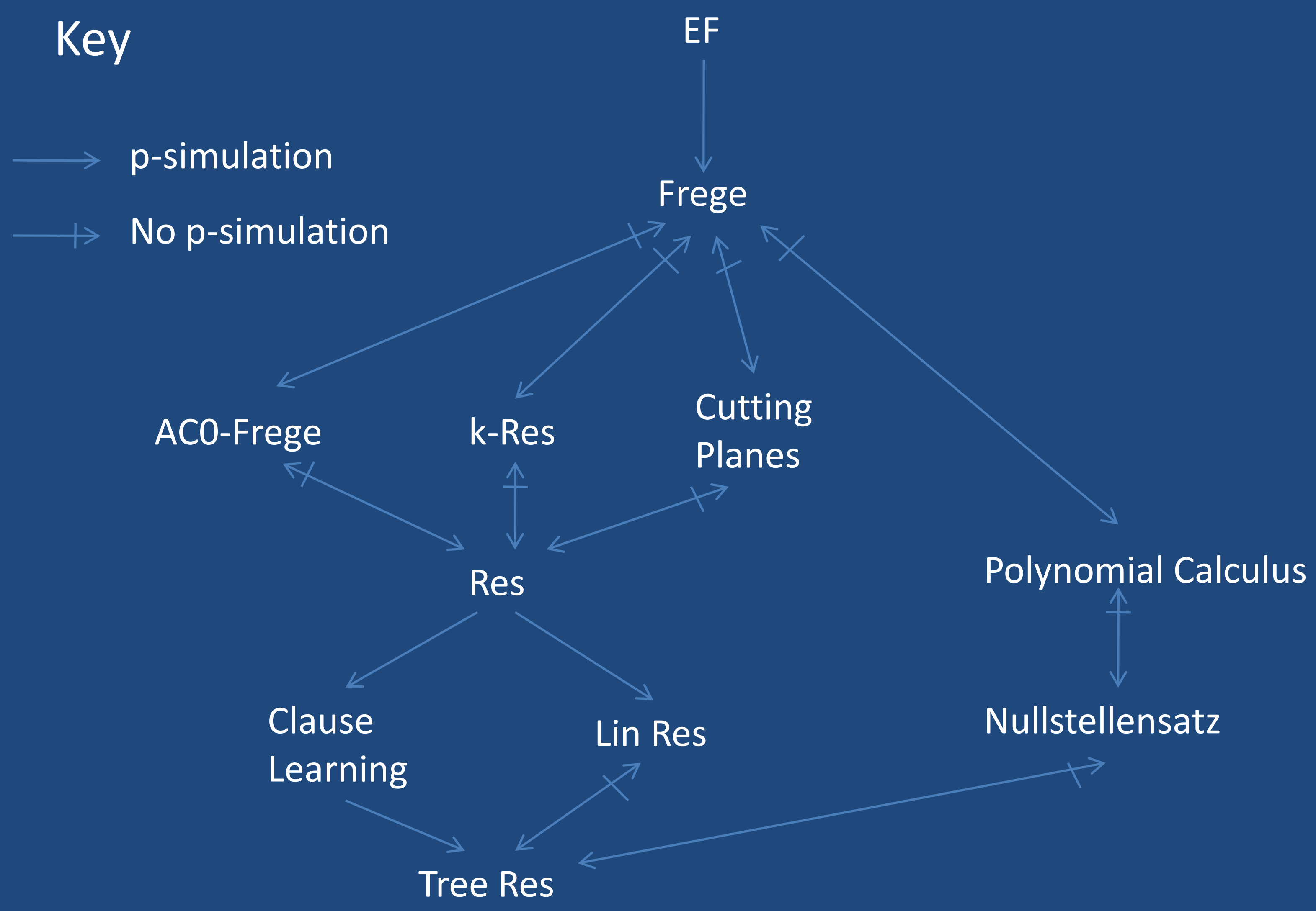
Proof Systems: Hilbert-style (Quantified)

- Axioms, rules of deduction, lines of proof are quantified Boolean formulae
- Key rule of deduction is *cut* rule (from $A \vee B \rightarrow C$ and $A \rightarrow B \wedge D$, derive $A \rightarrow D \vee C$)
- Different proof systems depending on type of B
  - B is Σ_i formula: G_i

Proof Systems: Algebraic

- Manipulating systems of polynomial equations: Polynomial Calculus (PC), Nullstellensatz
- Manipulating systems of linear inequalities: Cutting Planes (CP), Lovasz-Schrijver (LS), LS+

p-Simulations: The Map



Effectively p-simulations: Examples (1)

- Proposition: If A and B are (quasi)automatizable, then each effectively (quasi)p-simulates the other
- Corollary: Nullstellensatz, PC and Tree Resolution effectively (quasi)p-simulate each other
- Theorem [CEI96]: Nullstellensatz does not (quasi)p-simulate PC
- Tree Resolution does not (quasi)p-simulate Nullstellensatz or PC

Effectively p-simulations: Examples (2)

- Linear Resolution: Resolution where one of the resolved clauses is the most recently derived
- Unknown whether Linear Resolution p-simulates Resolution
- Theorem [B-OP03]: Linear Resolution effectively p-simulates Resolution

Effectively p-simulations: Examples (3)

- Clause Learning: Variant of Resolution used extensively in SAT solvers
- Unknown whether Clause Learning p-simulates Resolution
- Theorem [BHPvG08]: Clause Learning effectively p-simulates Resolution

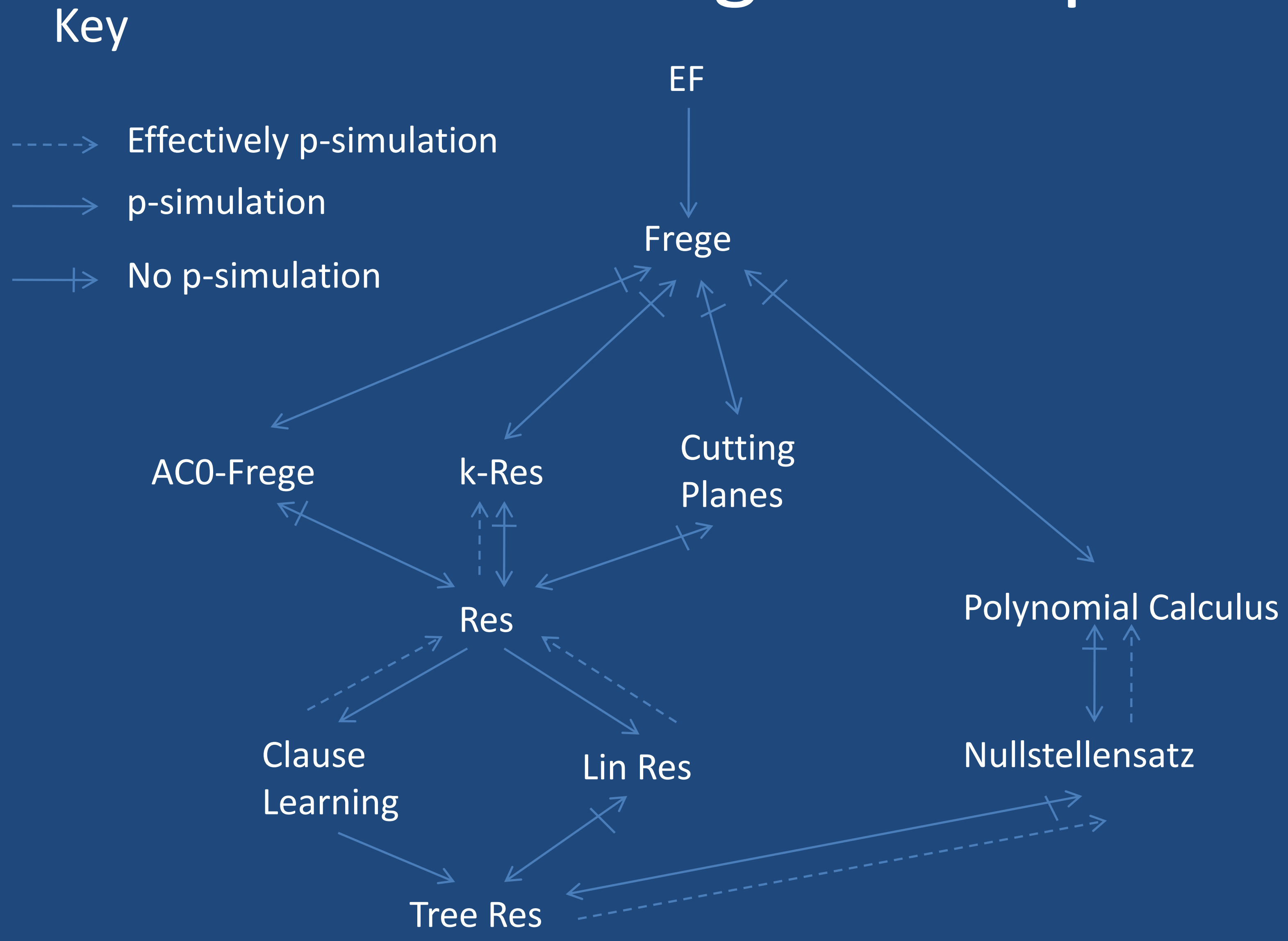
Effectively p-simulations: Examples (4)

- Theorem [ABE02]: Res does not p-simulate k-Res, for any $k >= 2$
- Theorem [AB04]: Res effectively p-simulates k-Res for any constant k
- Generalization: a proof system can effectively p-simulate any *local extension* of it

Effectively p-simulations: Examples (5)

- Unknown whether G_i p-simulates G_j , for $j < i$
- Theorem: G_0 effectively p-simulates *every* quantified proof system S
- Proof idea: Map ϕ to $Refl_S \rightarrow \phi$, and prove that if ϕ has small proofs in S, then $Refl_S \rightarrow \phi$ has small proofs in G_0

Re-drawing the Map



Lower Bounds on Effectively p-simulations

- If A is automatizable and B is not, then B does not effectively p-simulate A
- Corollary: If Factoring is not in quasi-poly time, then Tree Resolution does not eff. p-sim EF
- But how about if neither A nor B is believed to be automatizable?

Lower Bounds (ctd)

- Theorem: If NP $\cap$ coNP $\not\subseteq$ i.o.P, then there are prop. proof systems A and B such that
  - A is not automatizable
  - B is not automatizable
  - A does not effectively p-simulate B
- Analogue of Ladner's Theorem for proof complexity

Lower Bounds on Restricted Simulations

- Theorem: If Frege does not p-simulate EF, then there is no *symmetric extensional* effectively p-simulation of EF by Frege
- Uses result of [Clote-Kranakis91] about

  “poly-symmetric” functions

Open Problems

- More examples of effective p-simulations?
- Resolution does not effectively p-simulate EF, under natural assumption?
- Frege does not effectively p-simulate EF, for oblivious p-simulations?