Optimal proof systems – a survey

Olaf Beyersdorff

School of Computing
University of Leeds, UK

Outline of this survey

Background from proof complexity

- Resolution
- Frege and beyond
- ...

Do optimal proof systems exist?

- Characterisations
- Sufficient and necessary conditions
- Positive results in different models
- Connections to first-order logic
- Speculations

Background from proof complexity

Proof Systems

Definition (Cook, Reckhow 79)

A proof system for a language L is a function f with $rng(f) = L$.
If $f(w) = x$, then w is called an f-proof of $x \in L$.

- correctness: $rng(f) \subseteq L$
- completeness: $L \subseteq rng(f)$
- efficiency: proofs should be easy to check,
  i.e. f should be easy to compute.

- Most research in proof complexity has studied propositional proof systems where $L = TAUT$.

The Most Studied Proof System: Resolution

- Introduced by Blake 1937, Davis & Putnam 1960, and Robinson 1965
- Resolution proofs operate with clauses.
- Refutation system
- only one rule

$$\frac{C \vee p \qquad D \vee \neg p}{C \vee D}$$

- many subsystems studied: tree-like, regular ...
- tight relation to SAT solving

Complexity of Resolution

First historical lower bound:

- Pigeonhole principle: $n+1$ pigeons cannot sit in n holes
- CNF formulation PHP_n^{n+1}

$$\bigvee_{j\in[n]} x_{i,j} \qquad \text{for all pigeons } i \in [n+1]$$

$$\neg x_{i_1,j} \vee \neg x_{i_2,j} \qquad \text{for all distinct } i_1, i_2 \in [n+1] \text{ and } j \in [n]$$

- PHP_n^{n+1} requires Resolution refutations of size $2^{\Omega(n)}$. [Haken 85]

Many strong lower bounds

- Combinatorial principles: ordering principle, ...
- Graph-theoretic principles: Tseitin formulas, pebbling ...
- Random 3-CNF's are hard for Resolution. [Beame et al. 98]

A Strong System: Frege

Axioms

$$p_1 \to (p_2 \to p_1)$$
$$(p_1 \to p_2) \to (p_1 \to (p_2 \to p_3)) \to (p_1 \to p_3)$$
$$p_1 \to p_1 \vee p_2$$
$$p_2 \to p_1 \vee p_2$$
$$(p_1 \to p_3) \to (p_2 \to p_3) \to (p_1 \vee p_2 \to p_3)$$
$$(p_1 \to p_2) \to (p_1 \to \neg p_2) \to \neg p_1$$
$$\neg\neg p_1 \to p_1$$
$$p_1 \wedge p_2 \to p_1$$
$$p_1 \wedge p_2 \to p_2$$
$$p_1 \to p_2 \to p_1 \wedge p_2$$

Modus Ponens

$$\frac{p_1 \qquad p_1 \to p_2}{p_2}$$

Frege Proofs

A Frege proof of a formula φ is a sequence

$$(\varphi_1, \ldots, \varphi_n = \varphi)$$

of propositional formulas such that for $i = 1, \ldots, n$:

- φ_i is a substitution instance of an axiom, or
- φ_i was derived by modus ponens from φ_j, φ_k with $j, k < i$.

Major open problem

Show non-trivial lower bounds on the size of Frege proofs.

Restrictions and Extensions of Frege Systems

Bounded-depth Frege

Allow only formulas of logical depth d in the proof for a given constant d.

Extended Frege *EF*

Abbreviations for complex formulas: $p \equiv \varphi$,
where p is a new propositional variable.

Frege systems with substitution *SF*

Substitution rule: $\dfrac{\varphi}{\sigma(\varphi)}$

for arbitrary substitutions σ

Extensions of *EF*

Let Φ be a polynomial-time computable set of tautologies.
$EF + \Phi$: Φ as axiom schemes

Reductions between Proof Systems

Definition (Cook, Reckhow 79, Krajíček, Pudlák 89)

Let f and g be proof systems for L.

- f simulates g, if for any g-proof w there is an f-proof w' of length $|w'| = |w|^{O(1)}$ s.t. $f(w') = g(w)$.
- If w' is computable from w in polynomial time, then f p-simulates g.
- f and g are (p-)equivalent if they (p-)simulate each other.

Definition (Krajíček, Pudlák 89)

A proof system f for L is (p)-optimal if f (p-)simulates every proof system for L.

Simulations Between Proof Systems

Theorem (Cook, Reckhow 79)

All Frege systems are polynomially equivalent.

Theorem (Krajíček, Pudlák 89)

Every proof system is simulated by a proof system of the form $EF + \Phi$.

Problem (Krajíček, Pudlák 89)

Do optimal proof systems exist?

The Propositional Sequent Calculus

- Historically one of the first and best analyzed proof systems [Gentzen 35]
- basic objects: sequents $\varphi_1, \ldots, \varphi_m \vdash \psi_1, \ldots, \psi_k$.
- Sequents of the form

$$A \vdash A, \quad 0 \vdash, \quad \vdash 1$$

  are called initial sequents.
- An *LK*-proof of a propositional formula φ is a derivation of the sequent

$$\vdash \varphi$$

  from initial sequents by the following rules.

Rules of *LK*

$$\frac{\Gamma \vdash \Delta}{A, \Gamma \vdash \Delta} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A} \quad \text{(weakening)}$$

$$\frac{\Gamma_1, A, B, \Gamma_2 \vdash \Delta}{\Gamma_1, B, A, \Gamma_2 \vdash \Delta} \qquad \frac{\Gamma \vdash \Delta_1, A, B, \Delta_2}{\Gamma \vdash \Delta_1, B, A, \Delta_2} \quad \text{(exchange)}$$

$$\frac{\Gamma_1, A, A, \Gamma_2 \vdash \Delta}{\Gamma_1, A, \Gamma_2 \vdash \Delta} \qquad \frac{\Gamma \vdash \Delta_1, A, A, \Delta_2}{\Gamma \vdash \Delta_1, A, \Delta_2} \quad \text{(contradiction)}$$

$$\frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta} \qquad \frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A} \quad (\neg \text{ introduction})$$

$$\frac{A, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} \qquad \frac{A, \Gamma \vdash \Delta}{B \wedge A, \Gamma \vdash \Delta} \qquad \frac{\Gamma \vdash \Delta, A \qquad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B} \quad (\wedge \text{ rules})$$

$$\frac{A, \Gamma \vdash \Delta \qquad B, \Gamma \vdash \Delta}{A \vee B, \Gamma \vdash \Delta} \qquad \frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, A \vee B} \qquad \frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, B \vee A} \quad (\vee \text{ rules})$$

$$\frac{\Gamma \vdash \Delta, A \qquad A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \quad \text{(cut rule)}$$

A robust proof system: Frege/LK

Proposition (Cook, Reckhow 79)

Frege systems and the propositional sequent calculus LK are polynomially equivalent.

Polynomially Bounded Proof Systems

Polynomial Bounds on Proofs

A proof system f for L is polynomially bounded if there exists a polynomial p such that every $x \in L$ has an f-proof of size $\leq p(|x|)$.

Theorem (Cook, Reckhow 79)

A language L has a polynomially bounded proof system if and only if $L \in \mathrm{NP}$.

For propositional proof systems

TAUT has a polynomially bounded proof system if and only if $\mathrm{NP} = \mathrm{coNP}$.

Cook's Programme

Separate NP from coNP (and hence P and NP) by showing super-polynomial lower bounds to the size of proofs in all propositional proof systems.

Progress in this programme

- Haken (1985): exponential lower bound to the proof size in Resolution for the pigeonhole principle
- Ajtai (1988): Super-polynomial lower bound for bounded-depth Frege systems (Improved by Beame, Impagliazzo, Krajíček, Pitassi, Pudlák, Woods)
- Lower bounds for algebraic and geometric proof systems:
  - Cutting Planes [Pudlák 97]
  - Polynomial Calculus [Razborov 98, . . .]
  - Nullstellensatz [Buss et al. 97] [Grigoriev 98]
  - OBDD proof systems [Krajíček 08] [Segerlind 08]

Techniques and Barriers

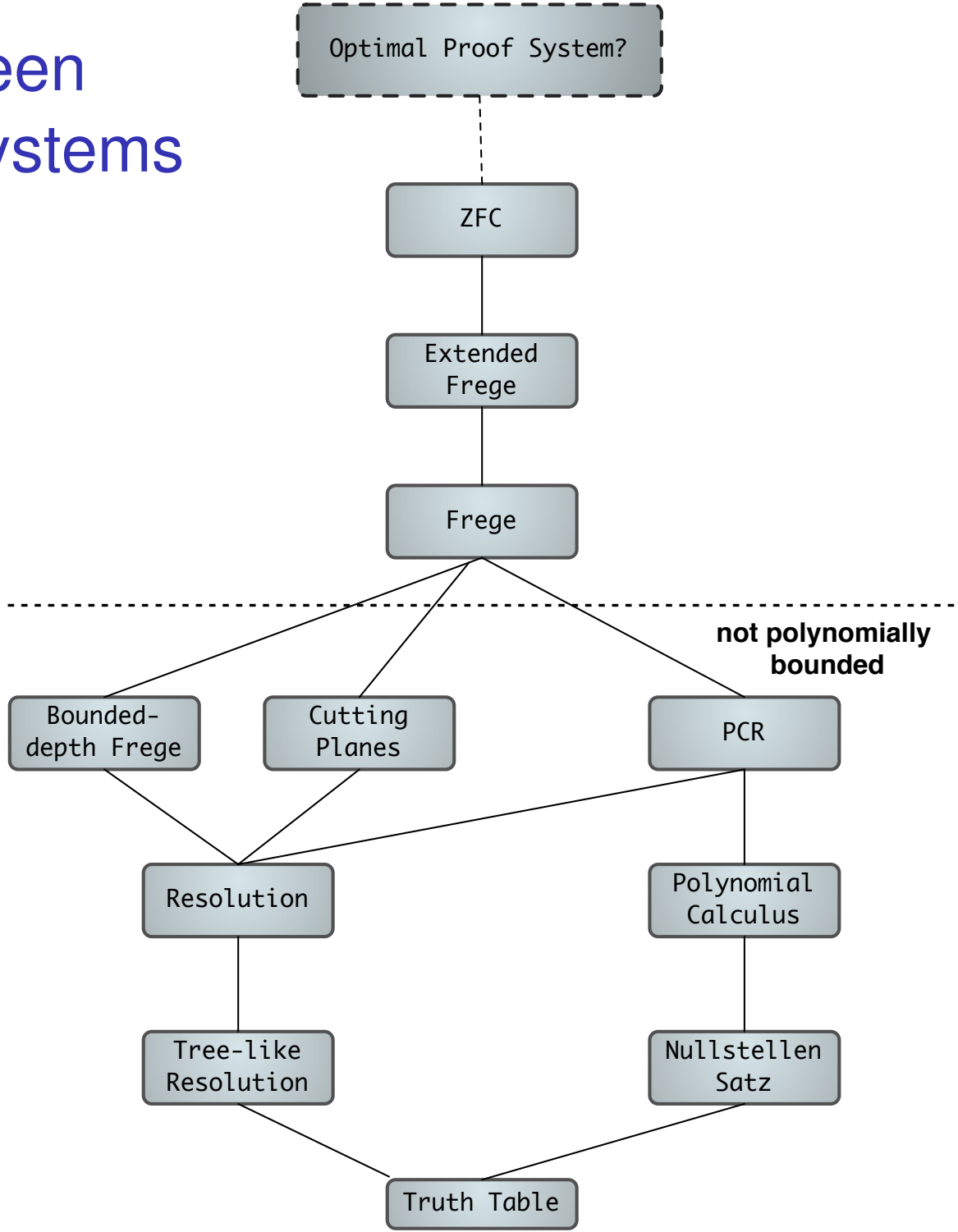
Techniques for lower bounds

- feasible interpolation [Krajíček 97]
- size-width relation [Ben-Sasson & Wigderson 01]
- game-theoretic techniques [Pudlák, Buss, Impagliazzo,. . .]
- proof complexity generators [Krajíček, Alekhnovich et al.]

The current barrier

Show lower bounds for Frege systems

Simulations between important proof systems



Characterisations

Optimal proof systems and acceptors

Definition

An optimal acceptor for L is a deterministic Turing machine M, which is optimal on the positive instances of L, i.e. M recognizes L and such that for every deterministic Turing machine M' which recognizes L there exists a polynomial p such that for every $x \in L$

$$time_M(x) \leq p(|x| + time_{M'}(x)).$$

Theorem (Krajíček & Pudlák 89)

TAUT has a p-optimal proof system if and only if TAUT has an optimal acceptor.

Generalized to

- SAT [Sadowski 99]
- and all paddable r.e. languages [Messner 99]

Optimal proof systems and easy subsets

Definition

A P-easy subset of L is a set $A \subseteq L$ with $L \in \mathrm{P}$.

Definition

A class $\mathbb{C}$ of languages has a recursive P-presentation if there exists a recursively enumerable list $N_1, N_2, \ldots$ of deterministic polynomial-time clocked Turing machines such that

- $L(N_i) \in \mathbb{C}$ for $i \in \mathbb{N}$ and
- for each $A \in \mathbb{C}$ there exists an index i with $A \subseteq L(N_i)$.

Optimal proof systems and easy subsets

Theorem (B & Sadowski 11)

Let L be a language such that the correctness of proof systems for L is expressible in L.
Then L has a p-optimal proof system if and only if
the P-easy subsets of L have a recursive P-presentation.

Theorem (Sadowski 02)

There exists a p-optimal propositional proof system if and only if the P-easy subsets of TAUT have a recursive P-presentation.

Similar characterisation for optimal proof systems

- replace P-presentation by NP-presentation
- you can also replace P-easy subsets by NP-easy subsets

Idea of proof "$\Rightarrow$"

- Let f be a p-optimal proof system for L and let A be a P-easy subset of L.
- Define a proof system f_A for L as follows:

$$f_A(x) = \begin{cases} f(y) & \text{if } x = 0y \\ a & \text{if } x = 1a \text{ and } a \in A \\ b & \text{otherwise} \end{cases}$$

  where b is a fixed element in L.
- Because f is p-optimal, f_A is p-simulated by f via some poly-time computable function t_A.
- Let $(t_i)_{i \in \mathbb{N}}$ be an enumeration of all deterministic poly-time clocked Turing transducers.
- For $i \in \mathbb{N}$ consider the following set of algorithms M_i:

```
1  Input:  x
2  IF f(t_i(1x)) = x THEN accept ELSE reject
```

P-optimal proof systems for SAT

Current knowledge

- All languages in NP have optimal proof systems.
- All languages in P have p-optimal proof systems.

Does SAT have a p-optimal proof systems?

- The standard proof system for SAT

$$sat(\alpha, \varphi) = \begin{cases} \varphi & \text{if } \alpha \text{ is a satisfying assignment for } \varphi \\ p & \text{otherwise.} \end{cases}$$

- Is sat p-optimal?

Is the standard proof system for SAT p-optimal?

Characterisations (B, Messner, Köbler 09)

The following are equivalent:

- The standard proof system for SAT is p-optimal.
- A number of complexity assumptions named Q [Fenner, Fortnow, Naik, Rogers 03]
- For all languages the notions of simulation and p-simulation coincide.
- Simulation and p-simulation coincide for propositional proof systems.
- Every optimal proof system is p-optimal.

Further Relations

Chen & Flum (2012) show surprising relations to

- descriptive complexity
- parameterized complexity
- main link through enumerations

Do optimal proof systems exist?

Necessary and sufficient conditions

Sufficient conditions

Theorem (Krajiček, Pudlák 89)

- *If* $\mathsf{E} = \mathsf{NE}$*, then TAUT has p-optimal proof systems.*
- *If* $\mathsf{NE} = \mathsf{coNE}$*, then TAUT has optimal proof systems.*

Theorem (Köbler, Messner, Torán 03)

Weakened assumptions to double exponential time:

- *If* $\mathsf{NEE} \cap \mathsf{Tally} \subseteq \mathsf{EE}$*, then TAUT has p-optimal proof systems.*
- *If* $\mathsf{NEE} \cap \mathsf{Tally} \subseteq \mathsf{coNEE}$*, then TAUT has optimal proof systems.*

Necessary conditions

Theorem (Köbler, Messner, Torán 03)

Optimal proof systems for TAUT imply complete sets for promise classes, e.g. NP ∩ Sparse*,* UP*, disjoint* NP*-pairs.*

Semantic Complexity Classes

For semantic classes there is no easy test verifying the correctness of the machine.

Examples

- UP $= \{L(M) \mid M$ is a nondeterministic polynomial-time Turing machine which on every input has at most one accepting path $\}$
- randomized classes: BPP, RP, ...
- NP ∩ coNP
- disjoint NP pairs

Complete sets for semantic complexity classes

Long searched for

NP ∩ coNP	[Kowalczyk 84]
UP	[Hartmanis, Hemachandra 88]
disjoint NP pairs	[Glaßer, Selman, Sengupta, Zhang 04] [B 07]

Theorem (Köbler, Messner, Torán 03)

Optimal proof systems imply complete sets for promise classes.

Could be interpreted as negative evidence

for the existence of optimal proof systems, but . . .

Theorem (Itsykson 10)

The average-case version of BPP has a complete problem.

Proof sketch

Theorem (Köbler, Messner, Torán 03)

Optimal proof systems for TAUT imply complete disjoint NP*-pairs.*

Proof idea

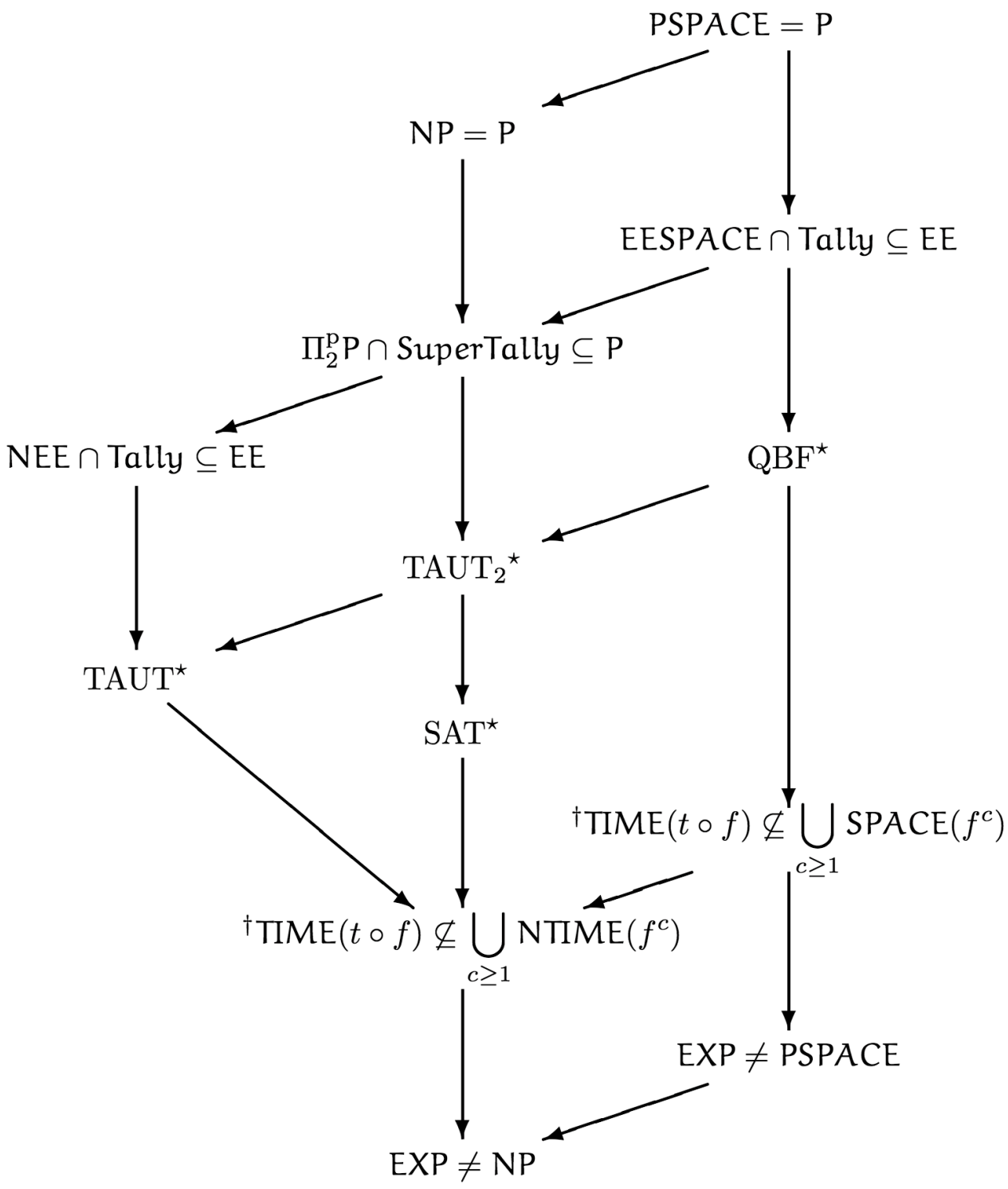
- Let (A, B) be a disjoint NP-pair.
- Express the disjointness of (A, B) by propositional tautologies

$$\neg\varphi_n(\bar{x}, \bar{y}) \vee \neg\psi_n(\bar{x}, \bar{z})$$

- these have short proofs in the optimal proof system
- use these proofs to verify the promise of the disjointness of two NP machines
- if promise holds, then simulate to obtain a complete set

Implications

L^* means that
L has p-optimal
proof systems



Source:
Messner 2001

Do optimal proof systems exist?

Positive results in different models

Proof systems that take advice

Cook & Krajiček (JSL 07) consider non-uniform Frege proofs.

Definition (Karp, Lipton 80)

- An advice function is a mapping $h : \mathbb{N} \to \Sigma^*$.
- $h(n)$ is the advice string provided by h for input length n.
- For a language L, $L/h = \{x \mid \langle x, h(|x|)\rangle \in L\}$.
- For a complexity class C and a length bound $k : \mathbb{N} \to \mathbb{N}$, $\mathsf{C}/k = \{L/h \mid L \in \mathsf{C}, |h(n)| \leq k(n) \text{ for all } n\}$.
- $\mathsf{C}/\log = \bigcup\{\mathsf{C}/k \mid k(n) = O(\log n)\}$.
- $\mathsf{C}/\mathrm{poly} = \bigcup\{\mathsf{C}/k \mid k(n) = n^{O(1)}\}$.

Proposition (Pippenger 79)

$L \in \mathsf{P}/\mathrm{poly}$ *iff* L *has poly-size circuits.*

Verifying proofs with advice

Traditional Cook-Reckhow model

Verify proofs in poly-time.

Proof systems with advice

Verify proofs in poly-time with the help of advice.

Question

Which languages have proof systems?

Answer

without advice: all r.e. languages
with 1 bit of advice: all languages

Advice helps in proving languages

Question

Which languages have polynomially bounded proof systems?

Answer (B, Köbler, Müller 11)

	advice on proof	advice on formula
ps/poly	NP/poly	NP/poly
$ps/\log$	NIC[log, poly]	NP/log
$ps/1$	NIC[log, poly]	NP/1
$ps/0$	NP	

Strict inclusions between these classes

$\text{NP} \subsetneq \text{NP}/1 \subsetneq \text{NP}/\log \subsetneq \text{NIC}[\log, \text{poly}] \subsetneq \text{NP}/\text{poly}$

All languages have optimal proof systems with advice

Theorem (Cook, Krajíček 07, B, Köbler, Müller 11)

Every language L has an optimal proof system f in $\mathrm{FP}/1$*. In fact, f simulates all proof systems in* $\mathrm{FP}/\log$ *for L, and p-simulates all proof systems in* FP *for L.*

All languages have optimal proof systems with advice

Theorem

Every language L has an optimal proof system f in $\mathrm{FP}/1$.

Proof.

- Let $\langle \cdot, \ldots, \cdot \rangle$ be a polynomial-time computable tupling function on Σ^* which is length injective.
- f-proofs are of the form $w = \langle u, 1^T, 1^m \rangle$ with $u, T \in \Sigma^*$ and $m \in \mathbb{N}$.
- The advice bit $h(|w|)$ indicates whether the transducer T only outputs elements from L for inputs of length $|u|$.
- Now, if $h(|w|) = 1$ and $T(u)$ outputs y after at most m steps, then $f(w) = y$. Otherwise, $f(w) = \top$.
- If g is a proof system computed by a p-time transducer T, then f p-simulates g via the FP function $u \mapsto \langle u, 1^T, 1^{p(|u|)} \rangle$.

□

Proof systems with oracle access

Verify proofs

in poly-time with oracle access

Theorem (B, Köbler, Müller 11)

1. *Every propositional proof system with logarithmic advice is simulated by a propositional proof system computable in polynomial time with access to a sparse* NP*-oracle.*
2. *Converse also holds.*

Corollary

Let $L \in$ coNP*. Then there exists a proof system* f *for* L *which simulates every poly-time computable proof system for* L*.*
f *is computable in polynomial time under a sparse* NP*-oracle.*

Sparse NP-sets are weak

$TAUT \notin \mathrm{NP}^S$ with a sparse NP-oracle S, unless the polynomial hierarchy collapses to its second level. [Kadin 89]

Probablistic acceptors

Definition (Hirsch, Itsykson 10)

- A pair (D, L) is a distributional proving problem if D is a family of probability distributions D_n with $\mu_{D_n}(\overline{L} \cap \{0,1\}^n) = 1$.
- A heuristic acceptor for (D, L) is a randomized algorithm that always accepts inputs from L and accepts inputs from $\overline{L}$ only with small probability.

Theorem (Hirsch, Itsykson 10)

Let L be recursively enumerable and D be a polynomial-time samplable distribution.
Then there is an optimal heuristic acceptor for (D, L).

Open problem

Does the equivalence between optimal acceptors and p-optimal proof systems extend to the heuristic case?
This would give an optimal heuristic proof system.

Optimal QBF proof systems under weak simulations

Definition (Pitassi, Santhanam 10)

A proof system Q effectively-p simulates proof system P if there is a polynomial-time truth-preserving transformation $(\phi, 1^m) \mapsto \phi'$, such that when ϕ has a P-proof of size $\leq m$, then ϕ' has a Q-proof of size polynomial in $|\phi| + m$.

Note

This is essentially equivalent to the condition that the canonical pair of P is reducible to the canonical pair of Q.

Theorem (Pitassi, Santhanam 10)

For any i, G_0 effectively-p simulates any proof system for Σ_i^q quantified boolean formulas.

Do optimal proof systems exist?

Connections to first-order logic

Bounded Arithmetic

- first-order arithmetic theories
- weak subsystems of Peano arithmetic
- axiomatized by
  - a number of basic axioms describing the interplay of $+, \cdot, \leq, 0, 1, \ldots$ and
  - some controlled amount of induction

Most important examples

- $I\Delta_0$ (induction for all bounded formulas)
- PV (formalizes poly-time computations) [Cook 75]
- $S_2^1 \subseteq T_2^1 \subseteq S_2^2 \subseteq T_2^2 \subseteq \ldots \subseteq S_2 = T_2$ [Buss 86]

Uniform vs. Non-uniform Concepts

	Complexity	Logic
uniform	P, NP, coNP, ... Turing machines	arithmetic theories Π_1^b formulas
non-uniform	AC^0, P/poly, NP/poly, ... Boolean circuits	proof systems propositional formulas

Bounded arithmetic and propositional proof systems

The correspondence

An arithmetic theory T corresponds to a propositional proof system P if the following conditions are satisfied:

1. For $\varphi \in \Pi^b_1$, if $T \vdash (\forall x)\varphi$, then there are poly-size P-proofs of propositional translations of φ.
2. T proves the correctness of P, i.e. $T \vdash RFN(P)$.
3. If $T \vdash RFN(Q)$, then Q is simulated by P.

This last two items mean that

P is optimal from the point of view of T.

Example

S^1_2 corresponds to extended Frege EF,
so EF is optimal from the perspective of S^1_2.

Quantitative Gödel's incompleteness theorem

Finitistic consistency statements

For a theory T let

$$Con_T(n) = \forall y, |y| \leq n \to \neg Prf_T(y, \ulcorner 0 = 1 \urcorner).$$

Theorem (Krajíček, Pudlák 89)

There exists an optimal propositional proof system if and only if there exists a consistent theory $S \supseteq S^1_2$ with a poly-time set of axioms such that for every consistent theory $T \supseteq S^1_2$ with a poly-time set of axioms the sentences $Con_T(n)$ have poly-size S-proofs.

Hard tautologies and optimal proof systems

Definition

A sequence of tautologies φ_n is hard for a proof system P, if φ_n is constructible in polynomial time and P does not have poly-size proofs of φ_n.

Theorem (Krajíček 95)

For all proof systems $P \geq EF$ (closed under substitutions and modus ponens) the following are equivalent:

1. *There exists a sequence of tautologies hard for P.*
2. *The proof system P is not optimal.*

Hard formulas for P come from reflection principles

$$RFN(Q) = (\forall \pi)(\forall \varphi) Prf_Q(\pi, \varphi) \rightarrow Taut(\varphi)$$

where Q is a proof system stronger than P

Hard formulas for strong systems

Hard candidate formulas for EF

- choose $RFN(Q)$ for a proof system Q that you believe to be stronger than EF, e.g. QBF sequent systems
- problem: hard to analyse

Hard formulas may be hard to find

- Suitable complexity assumptions about one-way functions imply that hard formulas cannot be efficiently constructed for a given system P. [Krajíček 13]
- Stronger assumptions about proof complexity generators imply that optimal proof systems exist. [Krajíček 11]

Do optimal proof systems exist?

If EF is optimal

- There are formulas without poly-size EF proofs (unless NP=coNP).
- But we cannot construct these efficiently.
- Does this explain our current dilemma to find candidates for hard formulas in EF?

Is the canonical EF pair complete for all disjoint NP pairs?

- no complexity consequences known

Do optimal proof systems exist?

Might not see the answer soon

- Most researchers seem to believe in a negative answer.
- Confirming this would separate complexity classes.
- Positive answer would have interesting consequences (optimal problems for promise classes).

Interesting cross-relations to

- logic
- complexity
- ...