The Complexity of Logic

Paul Beame

Computer Science and Engineering
University of Washington

Outline

1. Logic Review
    - Propositional logic
    - First-Order logic
2. Proofs in Logic
    - Propositional – Truth Tables, Resolution, Frege Systems
    - First-Order
3. Applications of Logic
4. Complexity and Propositional Proofs
    - Finding a proof vs. checking a proof
    - Efficient propositional proof systems
    - Relationship to NP = co-NP
5. Proof Complexity Lower Bounds
    - What lower bounds show
    - Some counting tautologies
    - The bounds
    - Techniques
6. Related Complexity Problems
7. Future Research
8. Other Recent Projects
    - Communication complexity (Joan Lawry)
    - Branching program bounds (Rakesh Sinha, Jayram Thathachar)

Propositional Logic

atoms:

$$p,\ q,\ r,\ \cdots$$

connectives:

$$\neg,\ \wedge,\ \vee,\ \Rightarrow,\ \Leftrightarrow,\ \cdots$$

formulas: e.g.

$$((\neg p \vee q) \Leftrightarrow (p \Rightarrow q))$$
$$((\neg p \wedge q) \vee r)$$

Truth valuation:

$$v : \text{atoms} \rightarrow \{T, \bot\}$$
$$v : \text{formulas} \rightarrow \{T, \bot\}$$

Formula F is

- *satisfiable* iff for some valuation v, $v(F) = T$.
- a *tautology* iff for all valuations v, $v(F) = T$.

F is a tautology $\Leftrightarrow$ $\neg F$ is not satisfiable

First-Order Predicate Logic

variables:

$$x,\ y,\ z,\ \cdots$$

constant symbols:

$$c_1,\ c_2,\ c_3,\ \cdots$$

function symbols:

$$f,\ g,\ h,\ \cdots \qquad \text{e.g. } f(x,y) = x + y$$

predicate symbols:

$$P,\ Q,\ R,\ \cdots \qquad \text{e.g. } P(x,y):\ x < y$$

connectives:

$$\neg,\ \wedge,\ \vee,\ \Rightarrow,\ \Leftrightarrow,\ \cdots$$

quantifiers:

$$\forall \text{ (for all)},\ \exists \text{ (there exists)}$$

formulas: e.g.

$$\forall x \forall y \forall z((\text{less}(x,y) \wedge \text{less}(y,z)) \Rightarrow \text{less}(x,z))$$

$$\forall x \exists y(f(x) = y)$$

interpretation: (Domain; values for all constant, function, relation symbols)

valid: true under all interpretations

Proofs in Propositional Logic

Truth Tables

- try all valuations

Proof of $((p \Rightarrow q) \Leftrightarrow (\neg p \vee q))$

p	q	$(p \Rightarrow q)$	$\neg p$	$(\neg p \vee q)$	$((\neg p \vee q) \Leftrightarrow (p \Rightarrow q))$
$\bot$	$\bot$	T	T	T	T
$\bot$	T	T	T	T	T
T	$\bot$	$\bot$	$\bot$	$\bot$	T
T	T	T	$\bot$	T	T

Sound: Anything proved is a tautology

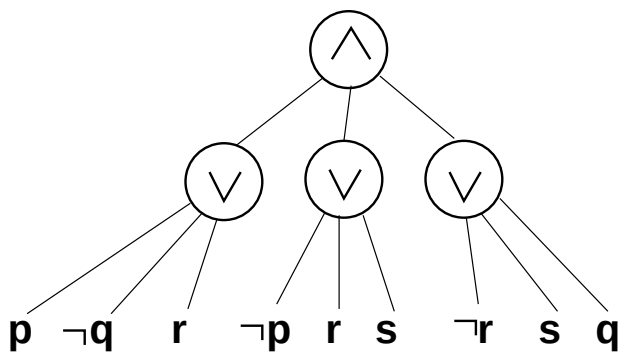
Complete: Every tautology has a proof

Propositional Proofs

Resolution:

- Shows formulas in *Conjunctive Normal Form (CNF)* not satisfiable
- "Refutation system"

CNF: $(p \vee \neg q \vee r) \wedge (\neg p \vee r \vee s) \wedge (\neg r \vee s \vee q)$



Clauses: $\{p, \neg q, r\}$, $\{\neg p, r, s\}$, $\{\neg r, s, q\}$

Resolution rule: Clauses $\{p\} \cup A$, $\{\neg p\} \cup B$ imply $A \cup B$

F not satisfiable $\Leftrightarrow$ empty clause derivable from F

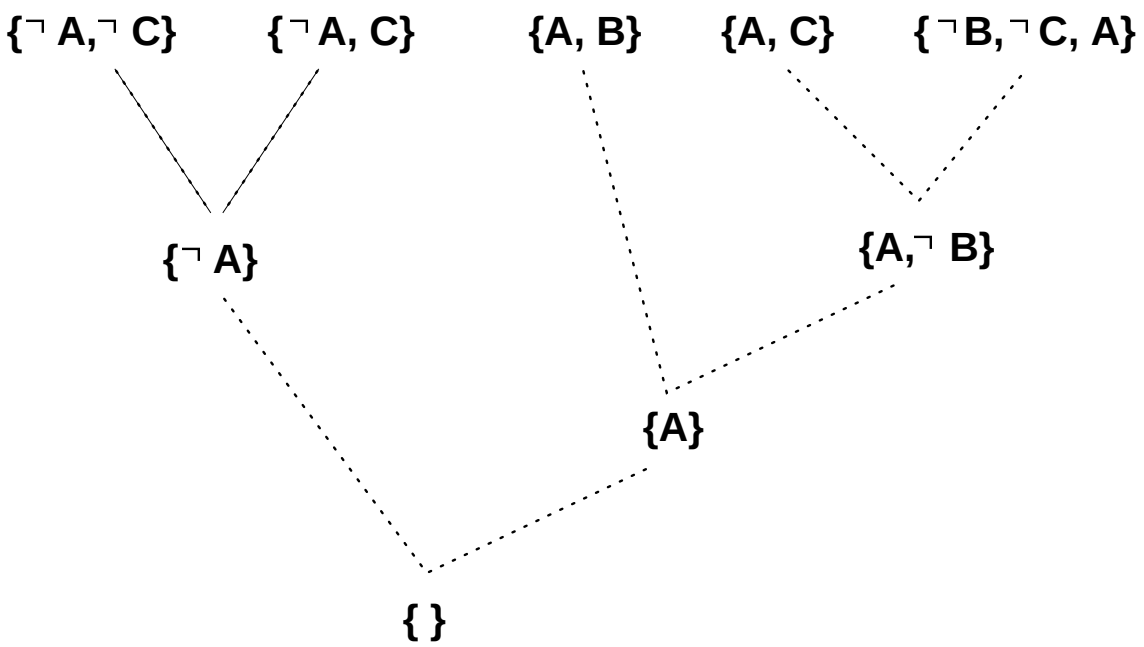
Completeness:

formula $F \hookrightarrow$ CNF formula F'
F tautology $\Leftrightarrow$ F' not satisfiable

Resolution Example

Refutation refutation of

$$((\neg A \vee \neg C) \wedge (\neg A \vee C) \wedge (A \vee B) \wedge (A \vee C) \wedge (\neg B \vee \neg C \vee A))$$



Propositional Proofs

Frege proofs:

- Finite number of axioms: e.g.

$$\frac{}{\therefore A \vee \neg A}$$

  (each a tautology)

- Finite number of inference rules: e.g.

$$\frac{A,\ A \Rightarrow B}{\therefore B}$$

$$\frac{A \vee C,\ B \vee \neg C}{\therefore A \vee B}$$

Can substitute aribtrary formulas into axioms/inference rules,
e.g. $(A \vee \neg A)$ gives $((p \Rightarrow q) \vee \neg(p \Rightarrow q))$

Proof of formula F: Sequence of formulas $F_1, \ldots, F_k$ such that

- Each F_i either
  - is an axiom, or
  - follows from previous F_j's via some inference rule
- F_k is F

Sound and Complete

Frege Proof example

Frege proof that $\neg p$ follows from $(p \Rightarrow (q \wedge \neg q))$:

1.	$p \Rightarrow (q \wedge \neg q)$	Given
2.	$\neg(q \wedge \neg q) \Rightarrow \neg p$	Contrapositive from 1
3.	$q \vee \neg q$	Excluded Middle
4.	$\neg(\neg q \wedge \neg\neg q)$	De Morgan's Law from 3
5.	$\neg(\neg q \wedge q)$	Double Negation from 4
6.	$\neg(q \wedge \neg q)$	Commutative Law from 5
7.	$\neg p$	Modus Ponens from 6 & 2

Predicate Logic Proofs

Frege Proofs: Add inference rules for each quantifier

e.g.

$$\frac{\forall x P(x)}{\therefore P(t) : \quad t \text{ is a term}}$$

$$\frac{P(t) : \quad t \text{ is a term}}{\therefore \exists x P(x)}$$

Resolution: Add "unification"

Sound and Complete

Applications of Proofs

AI

- Reasoning
- Planning, e.g. STRIPS-type planners

Program Verification

- Partial Correctness
- Propositional Dynamic Logic

Logic Programming – execution as proof

- Prolog

Circuit Verification

Automated theorem provers, e.g.

- Resolution Refutation System
- Boyer-Moore Theorem Prover
- NuPRL
- ML

Problems are often stated in Predicate Logic but operate on small finite structures so have equivalent Propositional Logic formulations

What is a Proof?

Properties of a proof:

- something you write down
- easy to verify correctness
- exists only for true formulas (tautologies)

May be hard to find

Formally ...

For a proof system S ...

there is a predicate Verify_S such that given P and formula F

- $\mathrm{Verify}_S(P, F)$ is true $\Leftrightarrow$ P is a proof of F
- Verify_S is computable in time polynomial in sizes of F and P.
- F is a tautology $\Leftrightarrow$ $\exists P.\ \mathrm{Verify}_S(P, F)$.

Proof system $S \equiv \mathrm{Verify}_S$ procedure

Efficiency of a Proof System

We know proofs are easy to check ... but often hard to find.

The real issue is how big they are ...

Define

$$\text{Complexity}_S(n) = \max_{F \text{ tautology}, |F| = n} \min\{|P| \ : \ \text{Verify}_S(P, F)\}$$

So

$$F \text{ is a tautology} \Leftrightarrow \exists P.\ |P| \leq \text{Complexity}_S(|F|).\ \text{Verify}_S(P, F)$$

Say proof system S is *efficient* iff Complexity_S is polynomial

e.g. $S =$ Truth Tables:

$$\text{Complexity}_S\ (n) = 2^{O(n)} \text{ – exponential, not efficient}$$

Some Complexity

- $L \in$ NP iff there is a polynomial-time computable predicate R such that

$$L = \{x \ : \ \exists y.\ |y| \leq |x|^{O(1)}.\ R(x,y)\}$$

- $L \in$ co-NP iff the complement of L, $\overline{L} \in$ NP

- $\mathrm{SAT} = \{F \mid F$ is a satisfiable propositional formula$\}$

- $\mathrm{TAUT} = \{F \mid F$ is a propositional tautology$\}$

$$= \{F \mid \exists P.\ |P| \leq \mathrm{Complexity}_S(|F|).\ \mathrm{Verify}_S(P,F)\}$$

for any propositional proof system S.

Verify_S is always polynomial-time computable and

S is efficient $\Leftrightarrow \mathrm{Complexity}_S(n) = n^{O(1)}$

So ...

an efficient propositional proof system exists $\Leftrightarrow \mathrm{TAUT} \in$ NP

The Heavy Guns

Recall . . .

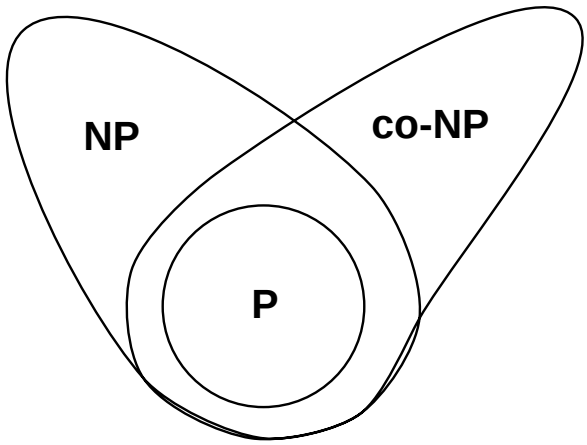
Cook's Theorem: SAT is NP-complete

Also . . . $F \in \mathrm{TAUT} \Leftrightarrow \neg F \notin \mathrm{SAT}$

Corollary: TAUT is co-NP-complete

So . . .

Efficient propositional proof systems exist $\Leftrightarrow$ $\mathrm{TAUT} \in$ NP
$\Leftrightarrow$ NP = co-NP



Classifying Propositional Proof Systems

Many proof systems in the logic literature

Cook and Reckhow (1979) classified their efficiency:

- Truth Tables
- Analytic Tableaux, Gentzen systems without cut rule
- Resolution
- Constant-depth Frege Systems – all equivalent
- Frege Systems, Gentzen Systems, Natural Deduction – all equivalent
- Extended Resolution, Extended Frege Systems, Frege Systems with substitution – all equivalent

Are any of these proof systems efficient?

Grand Goals

Show that no efficient propositional proof systems exist

$\Rightarrow$ NP $\neq$ co-NP

$\Rightarrow$ P $\neq$ NP

More modest goal:

Show that specific propositional proof systems people use are not efficient

Our job *should* be easier ...

Analyze a single nondeterministic algorithm for TAUT

vs

Analyze all possible nondeterministic algorithms for TAUT

Lower Bounds

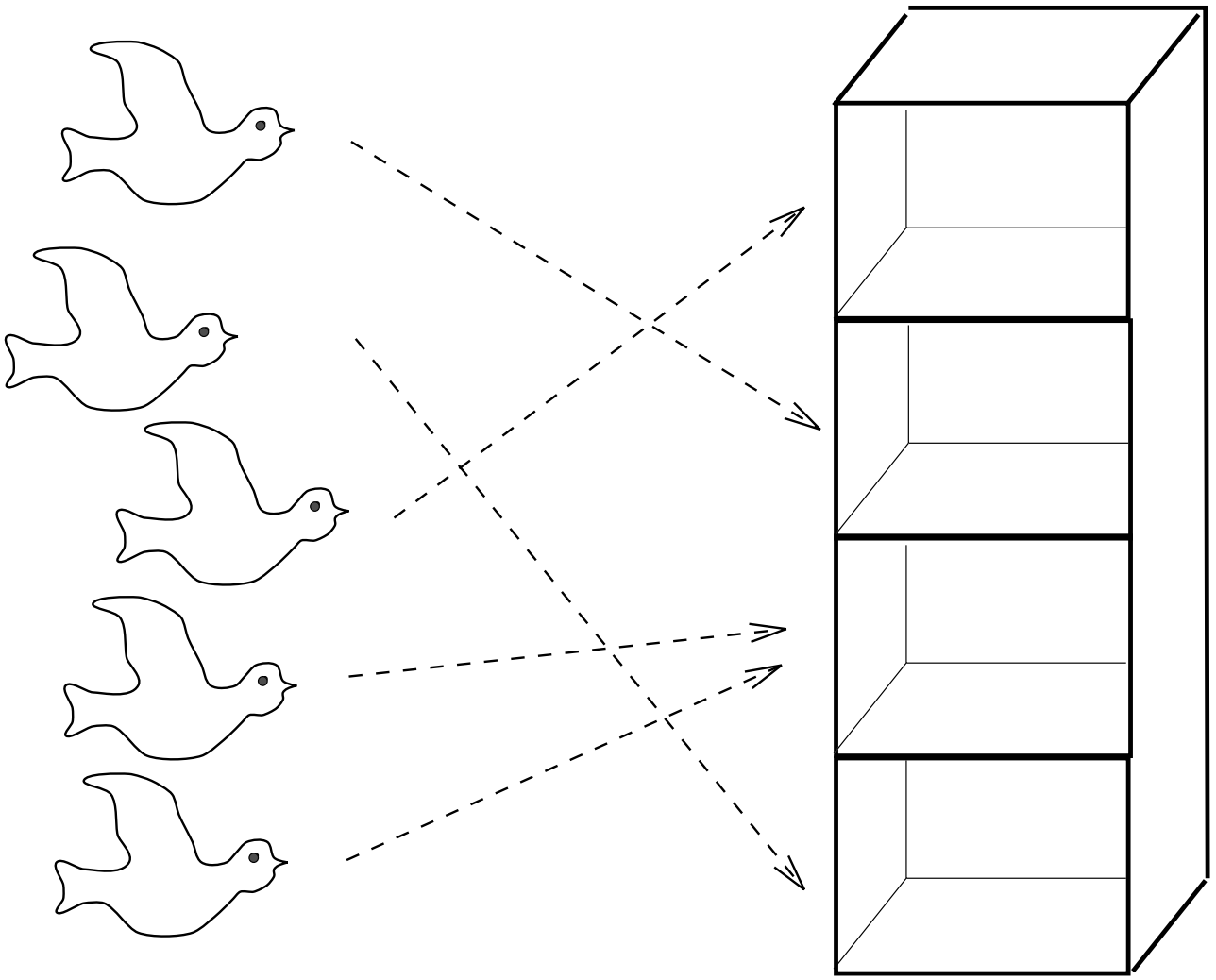
To show that proof system S is not efficient we find

- An infinite family of tautologies $\mathcal{F} = \{F_n\}$ such that $|F_{n+1}| > |F_n|$.
- An argument showing that for any polynomial function $p(\cdot)$, there is a formula F_n in $\mathcal{F}$ such that the shortest proof in S of F_n is bigger than $p(|F_n|)$

Such an argument is a called a $lower\ bound$

The families of tautologies we consider turn out to be natural counting principles

The Pigeonhole Principle



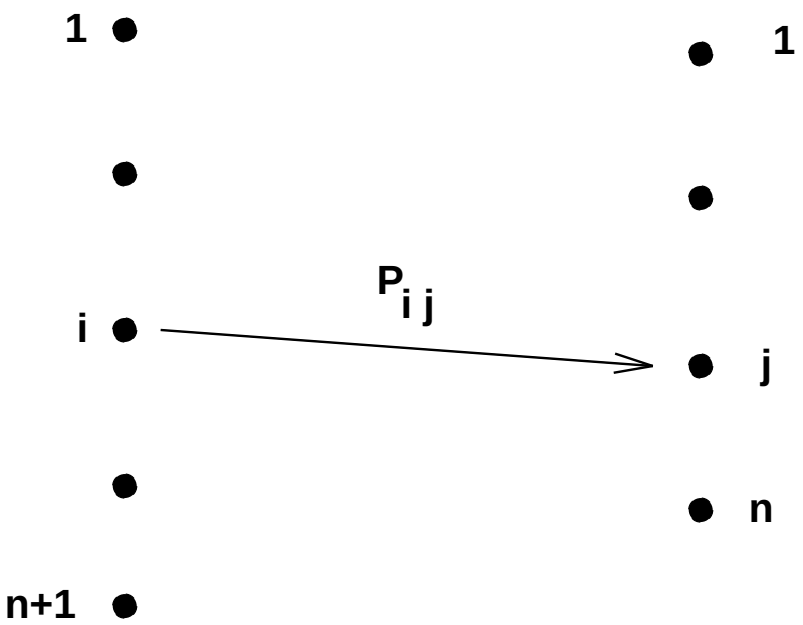
$n+1$
pigeons

n
pigeonholes

Any assignment of $n+1$ pigeons to n pigeonholes puts ≥ 2 pigeons in some pigeonhole

Essence of $<$

Pigeonhole Formula PHP_n



pigeons pigeonholes

Pigeon i is mapped to some pigeonhole:

$$(P_{i1} \vee P_{i2} \vee \cdots \vee P_{in})$$

Pigeons i and j are not both mapped to pigeonhole k

$$(\neg P_{ik} \vee \neg P_{jk})$$

$$\neg PHP_n = \bigwedge_{1 \leq i \leq n+1} (P_{i1} \vee P_{i2} \vee \cdots \vee P_{in}) \ \wedge \ (\bigwedge_{1 \leq k \leq n} \bigwedge_{1 \leq i < j \leq n+1} (\neg P_{ik} \vee \neg P_{jk}))$$

Landmark Lower Bounds

Tseitin (1968) "Regular" Resolution is not an efficient proof system

Galil (1975) "Regular" Resolution has exponential complexity

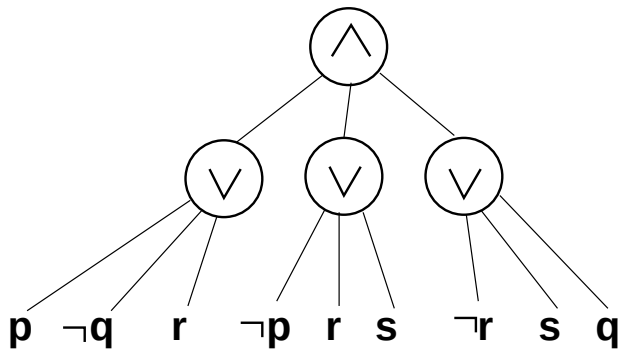
Haken (1984) PHP_n requires exponential size Resolution proofs

Chvatal & Szemeredi (1988) Almost all k-CNF formulas of cn clauses in n variables require exponential-size Resolution proofs

Ajtai (1988) PHP_n does not have efficient constant-depth Frege proofs

Lower Bounds

Constant-depth Frege Proofs:
Frege proofs only allowing formulas of fixed depth



Resolution is a special case: depth=2

Beame, Impagliazzo, Krajicek, Pitassi, Pudlak, Woods (1992)

Any constant-depth Frege proof of PHP_n requires exponential size

Any polynomial-size proof of PHP_n requires depth $\Omega(\log \log n)$.

Other Tautologies and Bounds

Mod_p^n tautologies: If p does not divide n then a set of size n cannot be evenly partitioned into groups of size p.

Ajtai (1989)

No efficient constant-depth Frege proofs of Mod_2^n even given PHP_n as axioms

Beame, Pitassi (1993)

Such proofs require exponential size

Ajtai (1993)
Beame, Impagliazzo, Krajicek, Pitassi, Pudlak (1994)
Riis (1994)

No efficient constant-depth Frege proofs of Mod_p given Mod_q as axioms

Technique

Borrowed from circuit complexity:

Assume there is some small constant-depth proof $F_1, \ldots, F_k$ of tautology F

Repeatedly simplify all formulas in the proof (including F)

Argue that

- simplied proof still "proves" simplified F
- simplifed F is still complicated
- simplified proof is too simple to "prove" a complicated formula

Contradiction

Related Complexity Problems

Papadimitriou (1990)

Search problems between P and NP

Given a polynomial-time Turing Machine description of a structure, find some object guaranteed to exist in the structure by famous math theorem.

e.g. Sperner's Lemma

Brouwer's Fixed Point Theorem

Borsuk-Ulam Theorem

Ham Sandwich Theorem

Classified based on the fundamental counting principle used

Beame, Cook, Edmonds, Impagliazzo, Pitassi (1994)

Classes are fundamentally different

Further Research

Improve Mod_p versus Mod_q separation to exponential

Find complexity of constant-depth Frege proofs of random k-CNF formulas

Study efficiency of more powerful proof systems

- Find candidate tautologies (counting tautologies too simple)
- Exploit other techniques from circuit complexity

Design new and interesting proof systems

Prove P $\neq$ NP (maybe this one is too hard)

Other Projects

Communication Complexity:

Beame, Joan Lawry (1992)

Randomized vs Nondeterministic Communication Complexity

- How many bits do two co-operating computers need to communicate to solve a problem when each has only part of the input?

Other Projects

Branching Program Bounds:

Rakesh Sinha, Jayram Thathachar (1994)

Efficient Oblivious Branching Programs for Threshold Functions

Branching Programs

- are like decision trees (remember those lower bounds for sorting), except that branches can come back together again
- model both Time and Space complexity at once
- generalizations of binary decision diagrams (b.d.d.'s) used in VLSI for verification