Proof Complexity

Lecture 1

Proof Systems (Computer Science & Math) Sep 3

What is a proof system? What is a proof?

- Mathematical proof
- "Proof by example"
- Formal methods – developing proofs with computer systems
- Proofs of termination and correctness of algorithms *
- Proofs of identity (crypto)
- Proof search – SAT solvers, linear programs

  primal $P \leftrightarrow$ dual P' (duality proofs)

- NP = {problems which have "efficiently verifiable solutions"} *

Mathematical Logic ~1900s

- Infinite objects – originally developed to formalize calculus
- Infinite objects have lots of problems

Gödel's Completeness Theorem *

If F is a true statement, then there is a (finite) proof of F.

(Cannot even be proved in ZF set theory.)

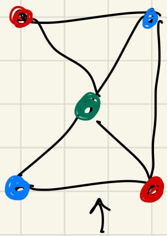
Computer science: "Everything" is finite!

- Solves foundational issues! "Simpler"
- Now: whether there exists short proofs of things!

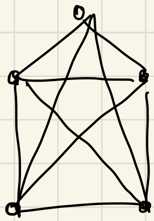
Computer Assisted Proofs

4-colour theorem

If G is a planar graph then we can colour the vertices of G s.t. each edge sees two different colours and we only use 4 distinct colours.



"planar" := draw without edge crossings



not planar

[Appel-Haken]

Proof: We reduce to ~2000 graphs.

Boolean Pythagorean Triples

Defn A pythagorean triple $(x,y,z) \in \mathbb{Z}^+$ s.t. $x^2+y^2=z^2$.

- infinitely many of them! $3^2+4^2=5^2$

Q. Can you colour all positive integers red or blue such that there are no monochromatic pythagorean triples?

[Heule - Kullman - Marek 16] No!

Thm $\{1, 2, \dots, 7824\}$ has a colouring w/out monochrome PTs. If you take $\{1, 2, \dots, 7825\}$ then it is impossible.

Proof := Produced by a SAT solver, original proof was 200 terabytes long, 80 gigabytes after compression.

2^{7825} two colourings $\rightsquigarrow$ ~ trillion colourings

Complexity of Proofs (in complexity) — Sep 3

- "Propositional" proofs, size of proofs
- Connections with

  Algorithms := proof search, optimization, boolean circuits

  Complexity := fundamental problems P vs. NP, optimization Unique Games Conjecture, proving circuit lower bounds, cryptography (zero-knowledge) etc.

Propositional Proof System (Complexity theory defn)

- $\{0,1\}^*$:= {set of all boolean strings}
- Let $L \subseteq \{0,1\}^*$ be a language (or decision problem)

Defn A proof system for a language L is a polynomial time algorithm V s.t.

$$\forall x \in \{0,1\}^* : x \in L \iff \exists p \in \{0,1\}^* : V(x,p) \text{ accepts.}$$

- The string p is called a proof
- Alg. V called verifier, verifies that p is a proof of x.

V is polynomially bounded if $\forall x \in L\ \exists p \in \{0,1\}^*$ $|p| \le poly(|x|)$ s.t. $V(x,p)$ accepts.

ex) SAT := {all satisfiable boolean formulas} ($\subseteq \underbrace{\{0,1\}^*}_{\text{encodings}}$) Sep 3

- Boolean formulas are composed of propositional variables $x_1, x_2, \cdots x_n \in \{0,1\}$ (False/True)

connected by connectives

AND := $\wedge$ OR := $\vee$ NOT := $\overline{\cdot}$

e.g. $F = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_2 \vee \overline{x_4}) \wedge (x_3 \vee x_4)$

F is <u>satisfiable</u> if there is an assignment that makes F evaluate to 1 (True).

Q. Polynomially-bounded proof system for SAT?

Proof := $p \in \{0,1\}^n$ – satisfying assignment

$V(F, p)$:= plugs p into the formula F, evaluates F, outputs Accept if F is satisfied.