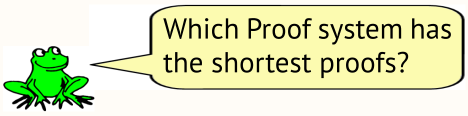
A Gentle Introduction to Deep Inference

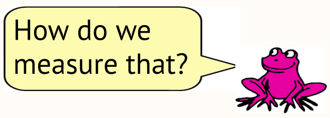
# 8. Lecture

# What is Proof Complexity?

Victoria Barrett and Lutz Straßburger








- Spoiler: We don't know which Proof system has the shortest proofs.
- But we know how to measure "the length of a proof".

## What is a proof system?

Notation: $\Sigma^*$ = set of all finite words over an alphabet $\Sigma$

**Definition:** Let $L \subseteq \Sigma^*$. A *proof system* for $L$ is a surjective function $f \colon \Upsilon^* \to L$, where $\Upsilon$ is another alphabet and $f$ is computable in polynomial time by a deterministic Turing machine (i.e., $f \in \mathbf{P}$).

Let $y \in L$. If $x \in \Upsilon^*$ and $y = f(x)$, then $x$ is a *proof* of $y$.

**Definition:** A proof system $f \colon \Upsilon^* \to L$ is *polynomially bounded* if there is a polynomial $p$ such that for all $y \in L$, there is a proof $x \in \Upsilon^*$ with $y = f(x)$ and $|x| \leq p(|y|)$.

**Theorem:** Let TAUT be the set of all Boolean tautologies. If there is a polynomially bound proof system for TAUT, then **coNP** = **NP**.



- Here $|z|$ is the length of the string $z$.
- **Exercise 8.1:** Prove the theorem (Hint: Note that SAT is **NP**-complete.)
- These definitions and theorems are due to
  - Stephen A. Cook and Robert A. Reckhow: **"The Relative Efficiency of Propositional Proof Systems".** *The Journal of Symboloc Logic 44(1), 1979*

## How to compare proof systems?

**Definition:** Let $f_1 \colon \Upsilon_1^* \to L$ and $f_2 \colon \Upsilon_2^* \to L$ be two proof systems for $L$. We say that $f_2$ *p-simulates* $f_1$ if there is function $g \colon \Upsilon_1^* \to \Upsilon_2^*$ such that $g \in \mathbf{P}$ and $f_2(g(x)) = f_1(x)$ for all $x \in \Upsilon_1^*$.

**Proposition:** If a proof system $f_2$ for $L$ p-simulates a proof system $f_1$ for $L$, and $f_1$ is polynomially bounded then $f_2$ is also polynomially bounded.

**Definition:** Two proof systems $f_1 \colon \Upsilon_1^* \to L$ and $f_2 \colon \Upsilon_2^* \to L$ are *p-equivalent* if $f_1$ p-simulates $f_2$ and $f_2$ p-simulates $f_1$.



- $g$ translates a proof $x$ of $y$ in the proof system $f_1$ into a proof $g(x)$ of $y$ in the proof system $f_2$.
- **Exercise 8.2:** Prove this Proposition.

## Frege Systems

*Axioms:*

$$A \to (B \to A) \qquad (A \wedge B) \to A$$
$$(A \to (B \to C)) \to (A \to B) \to A \to C \qquad (A \wedge B) \to B$$
$$A \to (A \vee B) \qquad A \to (B \to (A \wedge B))$$
$$B \to (A \vee B) \qquad \bot \to A$$
$$(A \to C) \to (B \to C) \to ((A \vee B) \to C) \qquad \neg\neg A \to A$$

*Rule:*

$$\text{mp}\,\frac{A \quad A \to B}{B}$$

Different Frege systems have different sets of axioms.

**Theorem:** All Frege systems are p-equivalent.



- Hilbert systems and Frege systems are the same.
- In proof theory they are usually called Hilbert systems, and in proof complexity they are called Frege systems.
- **Exercise 8.3:** Prove the theorem.

How do Frege systems compare to other proof systems (sequent calculus, deep inference, ...)