TAUT, TFNP and SAT

Pavel Pudlák
Mathematical Institute, Czech Academy of Sciences, Prague

Simons SAT Program Seminar, April 14, 2021

Fundamental theories

$T_2 \subset I\Sigma_1 \subset PA \subset SOA \subset ZFC \subset$

ZFC+strongly inaccessible cardinal $\subset$ ZFC+measurable cardinal ...

Fundamental theories

$T_2 \subset I\Sigma_1 \subset PA \subset SOA \subset ZFC \subset$

$ZFC+$strongly inaccessible cardinal $\subset ZFC+$measurable cardinal ...

By Gödel's Theorem they prove more true Π_1 sentences, but do they prove more sentences relevant to computational complexity?

Fundamental theories

$T_2 \subset I\Sigma_1 \subset PA \subset SOA \subset ZFC \subset$

$ZFC+$strongly inaccessible cardinal $\subset ZFC+$measurable cardinal ...

By Gödel's Theorem they prove more true Π_1 sentences, but do they prove more sentences relevant to computational complexity?

Related question:
Is the hierarchy of subtheories $T_2^1 \subseteq T_2^1 \subseteq \ldots$ of Bounded Arithmetic strictly increasing?

Example: proof complexity of algorithms

- Given a problem P and an algorithm A that solves P, we can ask:

  What is the weakest theory that proves the soundness of A?

Example: proof complexity of algorithms

- Given a problem P and an algorithm A that solves P, we can ask:

  What is the weakest theory that proves the soundness of A?
- Given a problem P and a time bound t, we can ask:

  What is the weakest theory T such that for some A, T proves that A solves P in time t?

Example: proof complexity of algorithms

- Given a problem P and an algorithm A that solves P, we can ask:

  What is the weakest theory that proves the soundness of A?

- Given a problem P and a time bound t, we can ask:

  What is the weakest theory T such that for some A, T proves that A solves P in time t?

Example (KP'94)

If FACTORING is hard, then S^1_2 does not prove the soundness of any polynomial time algorithm for PRIMALITY.

Example: proof complexity of algorithms

- Given a problem P and an algorithm A that solves P, we can ask:

  What is the weakest theory that proves the soundness of A?
- Given a problem P and a time bound t, we can ask:

  What is the weakest theory T such that for some A, T proves that A solves P in time t?

Example (KP'94)

If FACTORING is hard, then S_2^1 does not prove the soundness of any polynomial time algorithm for PRIMALITY.

We are not able to prove the soundness of AKS algorithm in any fragment of T_2.

- The *soundness* of algorithm A for a problem P means that A solves P.

- The *soundness* of algorithm A for a problem P means that A solves P.
- We can always formalize A so that ZFC (or any theory) does not prove the soundness.

- The *soundness* of algorithm A for a problem P means that A solves P.
- We can always formalize A so that ZFC (or any theory) does not prove the soundness.
- Can we formalize every algorithm so that its soundness is provable in PA (or some other fixed theory)?

Syntactic and semantic classes

The rule of thumb:

- syntactic classes have complete problems
- semantic classes do not

Syntactic and semantic classes

The rule of thumb:

- syntactic classes have complete problems
- semantic classes do not

The reason why a semantic class $\mathcal{C}$ does not have complete problems is:

1. we need a proof of the defining condition to show $P \in \mathcal{C}$,
2. there is no single theory T that is able to prove it for all $P \in \mathcal{C}$.

TAUT

TAUT$=_{df}$ DNF tautologies

Proof systems

1. complete (can always be made syntactic)
2. sound (semantic)

Polynomial simulations

TAUT

TAUT$=_{df}$ DNF tautologies

Proof systems

1. complete (can always be made syntactic)
2. sound (semantic)

Polynomial simulations

Conjecture (TAUT conjecture)

Equivalent formulations

1. *There is no proof system that simulates all proof systems.*
2. *There is no consistent theory that proves the soundness of all proof systems.*

TAUT

TAUT$=_{df}$ DNF tautologies

Proof systems

1. complete (can always be made syntactic)
2. sound (semantic)

Polynomial simulations

Conjecture (TAUT conjecture)

Equivalent formulations

1. *There is no proof system that simulates all proof systems.*
2. *There is no consistent theory that proves the soundness of all proof systems.*

Proposition

TAUT *conjecture* $\rightarrow$ EXP$\neq$NEXP.

DisjNP

DisjNP$=_{df}$ $\{(A, B) \mid A, B \in \mathrm{NP} \wedge A \cap B = \emptyset\}$

Polynomial reductions (Turing or many-one)

DisjNP

DisjNP$=_{df}$ $\{(A, B) |\ A, B \in \mathrm{NP} \wedge A \cap B = \emptyset\}$

Polynomial reductions (Turing or many-one)

Conjecture (DisjNP conjecture)

Equivalent formulations

1. *There is no complete disjoint* **NP** *pair.*
2. *There is no consistent theory that proves the disjointness of all disjoint* **NP** *pairs.*

The canonical pair of a proof system P

Definition (R'94)

$A_P = \{(\phi, 0^n) |\ \phi \in CNF \wedge \exists\ P$-refutation of ϕ of length $\leq n\}$;

$SAT^* = \{(\phi, 0^n) |\ \phi \in SAT\}$.

The canonical pair of a proof system P

Definition (R'94)

$A_P = \{(\phi, 0^n) | \ \phi \in CNF \wedge \exists \ P\text{-refutation of } \phi \text{ of length } \leq n\};$

$SAT^* = \{(\phi, 0^n) | \ \phi \in SAT\}.$

Fact

If P simulates Q, then (A_Q, SAT^*) *is reducible to* (A_P, SAT^*).

Corollary (KMT'03)

DisjNP *conjecture* $\Rightarrow$ TAUT *conjecture.*

TFNP

TFNP = Total Function NP

Definition

A TFNP problem is given by a binary relation R in **P** and a polynomial bound r such that

$$\forall a \exists b \; |b| \leq r(|a|) \wedge R(a, b).$$

The task is, for a given a, to find b such that $|b| \leq r(|a|) \wedge R(a, b)$.

TFNP

TFNP = Total Function NP

Definition

A TFNP problem is given by a binary relation R in **P** and a polynomial bound r such that

$$\forall a \exists b\ |b| \leq r(|a|) \wedge R(a, b).$$

The task is, for a given a, to find b such that $|b| \leq r(|a|) \wedge R(a, b)$.

Reduction R to R'

- many-one: $R'(f(a), b) \rightarrow\ R(a, g(a, b))$,
- or Turing: $R(a, g^{\text{oracle}R'}(a))$

Example

FACTORING $\in$ TFNP. *We believe it is not solvable in polynomial time.*

Example

FACTORING $\in$ TFNP. *We believe it is not solvable in polynomial time.*

Conjecture (TFNP conjecture)

Equivalent formulations

1. *There is no complete TFNP problem.*
2. *There is no consistent theory that proves totality of all TFNP problems.*

Example

FACTORING $\in$ TFNP. *We believe it is not solvable in polynomial time.*

Conjecture (TFNP conjecture)

Equivalent formulations

1. *There is no complete TFNP problem.*
2. *There is no consistent theory that proves totality of all TFNP problems.*

Evidence?

- The set of provably total computable functions increases with the strength of the theories.
- The well-known characterizations of provably total TFNP problems in fragments of bounded arithmetic suggest that these sets also increase.

DisjCoNP

DisjCoNP$=_{df}$ $\{(A, B) |\ A, B \in \text{coNP} \wedge A \cap B = \emptyset\}$

Polynomial reductions (Turing or many-one)

DisjCoNP

DisjCoNP$=_{df}$ $\{(A, B) | \ A, B \in \text{coNP} \wedge A \cap B = \emptyset\}$

Polynomial reductions (Turing or many-one)

Conjecture (DisjCoNP conjecture)

Equivalent formulations

1. *There is no complete disjoint* **coNP** *pair.*
2. *There is no consistent theory that proves the disjointness of all disjoint* **coNP** *pairs.*

DisjCoNP

DisjCoNP$=_{df}$ $\{(A, B) |\ A, B \in \text{coNP} \wedge A \cap B = \emptyset\}$

Polynomial reductions (Turing or many-one)

Conjecture (DisjCoNP conjecture)

Equivalent formulations

1. *There is no complete disjoint* **coNP** *pair.*
2. *There is no consistent theory that proves the disjointness of all disjoint* **coNP** *pairs.*

Proposition

DisjCoNP *conjecture* $\Rightarrow$ TFNP *conjecture.*

$$
\begin{tikzcd}
\mathsf{DisjNP} \arrow[dr] & & & & \mathsf{DisjCoNP} \arrow[dl] \\
& \mathsf{TAUT} \arrow[dr] & & \mathsf{TFNP} \arrow[dl] & \\
& & \mathsf{P} \neq \mathsf{NP} & &
\end{tikzcd}
$$

"X" means: "X does not have a complete problem"

$$
\begin{tikzcd}
\mathsf{DisjNP} \arrow[dr] & & & & \mathsf{DisjCoNP} \arrow[dl] \\
& \mathsf{TAUT} \arrow[dr] & & \mathsf{TFNP} \arrow[dl] & \\
& & \mathsf{P} \neq \mathsf{NP} & &
\end{tikzcd}
$$

"X" means: "X does not have a complete problem"

Where is SAT?

SAT

As for TAUT, we have

- proof systems for SAT
- polynomial simulations

SAT

As for TAUT, we have

- proof systems for SAT
- polynomial simulations

But

- the standard proof system for SAT = satisfying assignments
- the standard proof system is polynomially bounded
- yet, some proof systems for SAT are not polynomially bounded

Example

Define a proof system $P^{FACTORING}$ *for* SAT *by defining a proof of* $\phi(\bar{x})$ *to be either*

1. *a satisfying assignment* $\bar{a}$*, or*
2. n *if* n *is a non-prime and* $\phi(\bar{x})$ *expresses the fact that* $\bar{x}$ *is a proper divisor of* n*.*

Example

Define a proof system $P^{FACTORING}$ *for* SAT *by defining a proof of* $\phi(\bar{x})$ *to be either*

1. *a satisfying assignment* $\bar{a}$*, or*
2. n *if* n *is a non-prime and* $\phi(\bar{x})$ *expresses the fact that* $\bar{x}$ *is a proper divisor of* n*.*

Fact

If FACTORING *is hard, then the standard proof system does not polynomially simulate this system.*

Some natural proof systems for SAT

Observation

$\phi(\bar{x}) \in$SAT iff $\exists \bar{x} \phi(\bar{x})$ is a quantified propositional tautology.

Some natural proof systems for SAT

Observation

$\phi(\bar{x}) \in$SAT iff $\exists \bar{x} \phi(\bar{x})$ is a quantified propositional tautology.

- G is a sequent calculus for quantified propositional tautologies.
- G_i is G restricted to Σ_i^q sequents.
- G_i^* is the tree-like version of G_i.
- G_1^* is polynomially equivalent to Frege systems w.r.t. propositional tautologies.

Some natural proof systems for SAT

Observation

$\phi(\bar{x}) \in$SAT iff $\exists \bar{x} \phi(\bar{x})$ is a quantified propositional tautology.

- G is a sequent calculus for quantified propositional tautologies.
- G_i is G restricted to Σ_i^q sequents.
- G_i^* is the tree-like version of G_i.
- G_1^* is polynomially equivalent to Frege systems w.r.t. propositional tautologies.

Proposition

The standard proof system polynomially simulates G_1^* w.r.t. existentially quantified propositions.

Proposition

The standard proof system polynomially simulates G_1^* w.r.t. existentially quantified propositions.

Proposition

The standard proof system polynomially simulates G_1^* w.r.t. existentially quantified propositions.

Theorem (witnessing for G_1^*, Cook 2002)

Given a G_1^*-proof of $\exists\bar{y}.\phi(\bar{x},\bar{y})$ and an assignment $\bar{x} := \bar{a}$, one can construct in polynomial time $\bar{b}$ such that $\phi(\bar{a},\bar{b})$ is true.

Proof of Proposition.

Given a proof of $\exists\bar{y}.\phi(\bar{y})$ we get in polynomial time $\bar{b}$ that satisfies $\phi(\bar{y})$. $\square$

Theorem

If there is an optimal proof system for SAT*, then there exists a complete problem in* TFNP*.*

Proof.

Given a TFNP problem R, we define a proof system P^R for SAT:

- same construction as with NONPRIME, i.e., a is a proof of satisfiability of $R(a, y)$.

Theorem

If there is an optimal proof system for SAT*, then there exists a complete problem in* TFNP*.*

Proof.

Given a TFNP problem R, we define a proof system P^R for SAT:

- same construction as with NONPRIME, i.e., a is a proof of satisfiability of $R(a, y)$.

Given a proof system Q for SAT, define a TFNP problem R^P:

- $R^Q(x, y)$ iff
    1. $x = (\phi, v)$, v is a Q-proof of ϕ, and y is a satisfying assignment for ϕ;
    2. $y = 0$ if x is not of this form.

Soundness of P implies that R^P is total.

Theorem

If there is an optimal proof system for SAT, *then there exists a complete problem in* TFNP.

Proof.

Given a TFNP problem R, we define a proof system P^R for SAT:

- same construction as with NONPRIME, i.e., a is a proof of satisfiability of $R(a, y)$.

Given a proof system Q for SAT, define a TFNP problem R^P:

- $R^Q(x, y)$ iff
    1. $x = (\phi, v)$, v is a Q-proof of ϕ, and y is a satisfying assignment for ϕ;
    2. $y = 0$ if x is not of this form.

Soundness of P implies that R^P is total.

- If P^R is reducible to Q, then R is reducible to R^Q.

Theorem

If there is an optimal proof system for SAT*, then there exists a complete problem in* TFNP*.*

Proof.

Given a TFNP problem R, we define a proof system P^R for SAT:

- same construction as with NONPRIME, i.e., a is a proof of satisfiability of $R(a, y)$.

Given a proof system Q for SAT, define a TFNP problem R^P:

- $R^Q(x, y)$ iff
    1. $x = (\phi, v)$, v is a Q-proof of ϕ, and y is a satisfying assignment for ϕ;
    2. $y = 0$ if x is not of this form.

Soundness of P implies that R^P is total.

- If P^R is reducible to Q, then R is reducible to R^Q.

Hence if Q is an optimal proof system for SAT, then R^Q is complete in TFNP.

□

Example

Suppose $P^{\text{FACTORING}}$ is reducible Q. Then, given a non-prime n, we get a Q-proof v of ϕ, where $\phi(\bar{x})$ expresses that x is a proper divisor of n.

Example

Suppose $P^{\mathsf{FACTORING}}$ is reducible Q. Then, given a non-prime n, we get a Q-proof v of ϕ, where $\phi(\bar{x})$ expresses that x is a proper divisor of n.

If b satisfies $R^Q((\phi, v), b)$, then b satisfies ϕ, hence it is a proper divisor of n.

Example

Suppose $P^{\text{FACTORING}}$ is reducible Q. Then, given a non-prime n, we get a Q-proof v of ϕ, where $\phi(\bar{x})$ expresses that x is a proper divisor of n.

If b satisfies $R^Q((\phi, v), b)$, then b satisfies ϕ, hence it is a proper divisor of n.

Hence we can compute a proper divisor of n using an oracle for solutions of R^Q.

Conjecture (SAT conjecture)

SAT *does not have an optimal proof system.*

Corollary

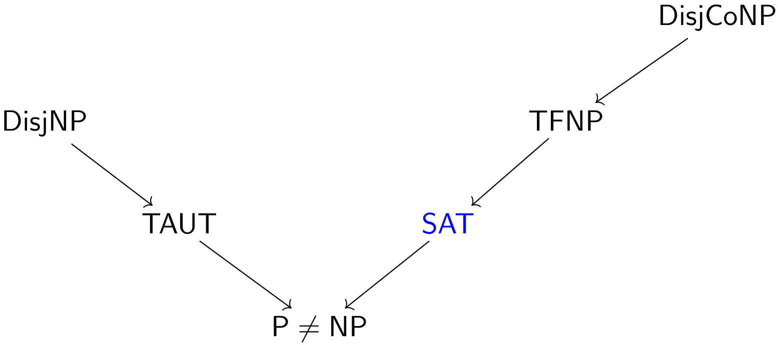
TFNP *conjecture* $\Rightarrow$ SAT *conjecture.*

Conjecture (SAT conjecture)

SAT *does not have an optimal proof system.*

Corollary

TFNP *conjecture* $\Rightarrow$ SAT *conjecture.*



"X" means: "X does not have a complete problem"

Relativizations

$$
\begin{tikzcd}
 & & & & & \mathrm{DisjCoNP} \arrow[dl, "?"'] \\
\mathrm{DisjNP} \arrow[dr, "\neq"] & & & & \mathrm{TFNP} \arrow[dl, "?"'] & \\
 & \mathrm{TAUT} \arrow[dr, "\neq"] & & \mathrm{SAT} \arrow[dl, "\neq"'] & & \\
 & & \mathrm{P} \neq \mathrm{NP} & & &
\end{tikzcd}
$$

- DisjCoNP $\not\Rightarrow$ TAUT [Khaniki'19]
- DisjNP $\not\Rightarrow$ SAT [Dose'20]

Thank You