Totality, Provability, and Feasibility

Sam Buss
U.C. San Diego

ASL Annual Meeting
Gödel Lecture
New York City
May 22, 2019
Topics:

- Formal theories of weak fragments of Peano arithmetic
  - First- and second-order theories of bounded arithmetic
- $\forall \exists$ consequences: Provably total functions
  - Computational complexity characterizations
- $\forall$ consequences: Universal statements
  - Cook translation to propositional logic
  - Paris-Wilkie translation to propositional logic

Underlying philosophy:

- A feasibly constructive proof that a function is total should provide a feasible method to compute it.
- A feasibly constructive proof of a universal statement should provide a feasible method to verify any given instance.
**Cook, 1975**, Feasibly constructive proofs and the propositional calculus

*...* A constructive proof of, say, a statement $\forall x A$ must provide an effective means of finding a proof of $A$ for each value of $x$, but nothing is said about how long this proof is as a function of $x$. If the function is exponential or super exponential, then for short values of $x$ the length of the proof of the instance of $A$ may exceed the number of electrons in the universe.

Introducing PV and the Cook translation
Parikh, 1971, Existence and feasibility in arithmetic

All this would tend to give some weight to the contentions that there is a definite concrete or anthropomorphic point of view possible in mathematics and that exponentiation would be excluded by this view as a genuine computable function. [...] We formulate a subsystem of Peano arithmetic which is related to the anthropomorphic viewpoint and investigate some of its properties.

Introducing $I\Delta_0$. 
First-/second-order theories of bounded arithmetic

\( \Pi_2 \)-consequences: Provably total functions

\( \Pi_1 \)-consequences: Translations to propositional logic
First-/second-order theories of bounded arithmetic

Computational complexity
Propositional proof complexity

$\Pi_2$-consequences: Provably total functions

$\Pi_1$-consequences: Translations to propositional logic
First-/second-order theories of bounded arithmetic

First-order theories of bounded arithmetic:
- \( \Pi_2 \)-consequences: Provably total functions
- \( \Pi_1 \)-consequences: Translations to propositional logic

Propositional proof complexity
Propositional proof search (SAT solvers)

Computational complexity
First-/second-order theories of bounded arithmetic

Computational complexity
Propositional proof complexity

Π₂-consequences: Provably total functions

Π₁-consequences: Translations to propositional logic

Propositional proof search (SAT solvers)
First-order theory $S^1_2$ of arithmetic:

- Terms have polynomial growth rate (smash, #, is used).
- Bounded quantifiers $\forall x \leq t$, $\exists x \leq t$.
- Sharply bounded quantifiers $\forall x \leq |t|$, $\exists x \leq |t|$, bound $x$ by $\log$ (or length) of $t$.
- Classes $\Sigma^b_i$ and $\Pi^b_i$ of formulas are defined by counting bounded quantifiers, ignoring sharply bounded quantifiers.
- $\Sigma^b_1$ formulas express exactly the NP predicates.
- $\Sigma^b_i$, $\Pi^b_i$ - express exactly the predicates at the $i$-th level of the polynomial time hierarchy.
- $S^1_2$ has polynomial induction PIND, equivalently length induction (LIND), for $\Sigma^b_1$ formulas $A$ (i.e., NP formulas):

$$A(0) \land (\forall x)(A(x) \rightarrow A(x+1)) \rightarrow (\forall x)A(|x|)$$
(1) Provably total functions of $S_2^1$:
   · The $\forall\Sigma^b_1$-definable functions (aka: \textit{provably total functions}) are precisely the polynomial time computable functions.
   · PV: equational theory over polynomial time functions. [C’75]
   · $S_2^1$(PV) is conservative over both $S_2^1$ and PV.

(2) Translation to propositional logic ("Cook translation")
   · Any polynomial identity ($\forall\Sigma^b_0$-property) provable in PV / $S_2^1$, has a natural translation to a family $F$ of propositional formulas. These formulas have polynomial size extended Frege ($e\mathcal{F}$) proofs.

(3) $S_2^1$ proves the consistency of $e\mathcal{F}$. Conversely, any propositional proof systems (p.p.s.) $S_2^1$ proves is consistent(provably) polynomially simulated by $e\mathcal{F}$.

(4) Lines (formulas) in an $e\mathcal{F}$ proof correspond to Boolean circuits. The circuit value problem is complete for $P$ (polynomial time).
Equational & First-order theories of bounded arithmetic

$\Pi_2$-consequences: Provably total functions

$\Pi_1$-consequences: Translations to propositional logic

Polynomial time functions ($P$)

extended Frege ($e\mathcal{F}$)
Proof lines are Boolean circuits (nonuniform $P$)

$PV / S^1_2$
Example of Cook translation: \( S^1_2, e\mathcal{F}, \text{PHP} \).

The first-order theory \( S^1_2 \) proves:

\[
(\forall x, n)[“\text{The bits of } x \text{ do not code an incidence matrix of a bipartite graph on } [n + 1] \cup [n] \text{ violating the Pigeonhole Principle PHP}^{n+1}_n”]
\]

Propositional translations \( \text{PHP}^{n+1}_n \): \((n \geq 1)\)

\[
\begin{align*}
\bigwedge_{i=0}^{n} \bigvee_{j=0}^{n-1} p_{i,j} & \quad \rightarrow \quad \bigvee_{i=0}^{n-1} \bigvee_{i'=i+1}^{n} \bigvee_{j=0}^{n-1} (p_{i,j} \land p_{i',j})
\end{align*}
\]

The propositional variables \( p_{i,j} \) correspond to the bits of the first-order variable \( x \).

Cook translation yields:

The \( \text{PHP}^{n+1}_n \) formulas have polynomial size \( e\mathcal{F} \) proofs. [CR]
Propositional proof systems ($\mathcal{F}$, $e\mathcal{F}$, ...)

**Frege proofs ($\mathcal{F}$):** Sequent calculus propositional system. Equivalent to a ‘textbook style’ proof system using modus ponens.

**Extended Frege proofs ($e\mathcal{F}$):** Frege systems augmented with extension rule allowing (iterated) introduction of new variables $x$ abbreviating formulas:

$$
\text{Extension axiom: } \quad x \leftrightarrow \varphi.
$$

**$\text{AC}^0$-Frege, aka constant-depth Frege:** Frege proofs over $\land, \lor, \neg$ with a constant bound on the number of alternations of $\land$’s and $\lor$’s. (Negations applied only to variables.)

**Quantified sequent calculus $\text{QBF}$ with $\forall p, \exists p$ Boolean quantifiers.** $G_i$ is $\text{QBF}$ restricted to $i$-levels of quantifiers.

Proof size $=$ number of symbols in the proof. (The purpose of extension is to reduce proof size.)
Open problems:

(1) Does the Frege system ($\mathcal{F}$) allow polynomial size proofs of tautologies? (Subexponential size?)

(2) Does the Frege system quasipolynomially simulate the extended Frege ($\mathcal{eF}$) system?
   
   · No good combinatorial candidates for separation are known. [BBP, HT, B, AB, ...]

(3) QBF versus $\mathcal{eF}$?
   
   · ($\mathcal{eF}$ is equivalent to $G_1^*$, i.e., tree-like $G_1$).
Theories for polynomial space

- PSA - Equational theory for PSPACE functions [D]
- $U_2^1$ - Second-order theory for polynomial space [B]
- The $\Sigma_1^{1,b}$-definable functions of $U_2^1$ are precisely the PSPACE functions.
- $U_2^1(PSA)$ is conservative over both $U_2^1$ and PSA. [**]
- PSPACE identities provable in $U_2^1$ have natural translations to QBF formulas which have polynomial size QBF proofs.
Weak second-order theories for weaker complexity [I,Z,...,CN]
These second-order theories use
(a) first-order objects playing the role of sharply bounded objects,
(b) second-order objects playing the role of inputs and outputs.
Base theory $V^0$ has comprehension and induction for bounded first-order formulas (with second order free variables).

Theories for ALogTime (uniform NC$^1$): [CT, A, CM, CN]
- Complexity class NC$^1$ - properties expressible by polynomial size Boolean formulas.
- VNC$^1$ - is $V^0$ plus axioms asserting the totality of the Boolean Formula Value Problem or log-bounded tree recursion. These are in NC$^1$ [B] and complete for NC$^1$.
- Provably total functions are precisely the functions of polynomial growth rate with NC$^1$ bit graph.
- Cook translation is to Frege proofs $\mathcal{F}$.
Theories for $L$ (log space) $[Z, P, CN]$
- $VL$ - is $V^0$ plus axioms asserting the totality of log-bounded recursion.
- Provably total functions are precisely the log-space computable functions.
- Cook translation is a tree-like p.p.s. $GL^*$ for $\Sigma$-$CNF(2)$ formulas, a class of QBF formulas complete for log space. [J]

Theories for $NL$ (nondeterministic log space) $[CK, P, CN]$
- $VNL$ - is $V^0$ plus axioms asserting the existence of a distance predicate for graph reachability.
- Provably total functions are precisely the polynomial growth rate functions with $NL$ bit graph.
- Cook translation is a tree-like p.p.s. $GNL^*$ for $\Sigma Krom$ formulas, a class of QBF formulas complete for $NL$. [G]

In progress: New p.p.s.’s $eLDT$ and $eLNDT$ for branching programs and nondeterministic branching programs as Cook translations for $VL$ and $VNL$. [B-Das-Knop, following Cook]
<table>
<thead>
<tr>
<th>Formal Theory</th>
<th>Propositional Proof System</th>
<th>Total Functions</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV, $S_2^1$, VPV</td>
<td>$\mathcal{F}$, $G_1^*$</td>
<td>$P$</td>
<td>[C, B, CN]</td>
</tr>
<tr>
<td>$T_2^1$, $S_2^2$</td>
<td>$G_1$, $G_2^*$</td>
<td>$\leq_{1-1}(\text{PLS})$</td>
<td>[B, KP, KT, BK]</td>
</tr>
<tr>
<td>$T_2^2$, $S_2^3$</td>
<td>$G_2$, $G_3^*$</td>
<td>$\leq_{1-1}(\text{CPLS})$</td>
<td>[B, KP, KT, KST]</td>
</tr>
<tr>
<td>$T_2^i$, $S_2^{i+1}$</td>
<td>$G_i$, $G_{i+1}^*$</td>
<td>$\leq_{1-1}(\text{LLI}_i)$</td>
<td>[B, KP, KT, KNT]</td>
</tr>
<tr>
<td>PSA, $U_2^1$, $W_1^1$</td>
<td>QBF **</td>
<td>$\text{PSPACE}$</td>
<td>[D, B, S]</td>
</tr>
<tr>
<td>$V_2^1$</td>
<td>**</td>
<td>$\text{EXPTIME}$</td>
<td>[B]</td>
</tr>
<tr>
<td>VNC$^1$</td>
<td>Frege ($\mathcal{F}$)</td>
<td>ALogTime</td>
<td>[CT, A; CM, CN]</td>
</tr>
<tr>
<td>VL</td>
<td>GL$^*$</td>
<td>L</td>
<td>[Z, P, CN]</td>
</tr>
<tr>
<td>VNL</td>
<td>GNL$^*$</td>
<td>NL</td>
<td>[CK, P, CN]</td>
</tr>
</tbody>
</table>

PV, PSA - equational theories.
$S_2^i$, $T_2^i$ - first order
$U_2^1$, $V_2^1$, VNC$^1$, VL, VNL, VPV - second order
<table>
<thead>
<tr>
<th>Formal Theory</th>
<th>Propositional Proof System</th>
<th>Total Functions</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV, $S^1_2$, VPV</td>
<td>$e\mathcal{F}, G^*_1$</td>
<td>P</td>
<td>[C, B, CN]</td>
</tr>
<tr>
<td>$T^1_2, S^2_2$</td>
<td>$G_1, G^*_2$</td>
<td>$\leq_{1-1}$(PLS)</td>
<td>[B, KP, KT, BK]</td>
</tr>
<tr>
<td>$T^2_2, S^3_2$</td>
<td>$G_2, G^*_3$</td>
<td>$\leq_{1-1}$(CPLS)</td>
<td>[B, KP, KT, KST]</td>
</tr>
<tr>
<td>$T^i_2, S^{i+1}_2$</td>
<td>$G_i, G^{*}_{i+1}$</td>
<td>$\leq_{1-1}$(LLI$_i$)</td>
<td>[B, KP, KT, KNT]</td>
</tr>
<tr>
<td>PSA, $U^1_2$, $W^1_1$, $V^1_2$</td>
<td>QBF</td>
<td>$PSPACE$</td>
<td>[D, B, S]</td>
</tr>
<tr>
<td></td>
<td>**</td>
<td>$EXPTIME$</td>
<td>[B]</td>
</tr>
<tr>
<td>VNC$^1$</td>
<td>Frege ($\mathcal{F}$)</td>
<td>ALogTime</td>
<td>[CT, A; CM, CN]</td>
</tr>
<tr>
<td>VL</td>
<td>$GL^*$</td>
<td>L</td>
<td>[Z, P, CN]</td>
</tr>
<tr>
<td>VNL</td>
<td>$GNL^*$</td>
<td>NL</td>
<td>[CK, P, CN]</td>
</tr>
</tbody>
</table>

Using Cook translation to propositional proof systems (p.p.s.’s)

$\mathcal{F}, e\mathcal{F}$ - Frege and extended Frege.

$G_i, QBF$ - quantified propositional logics.

Starred (*) propositional systems are tree-like.
<table>
<thead>
<tr>
<th>Formal Theory</th>
<th>Propositional Proof System</th>
<th>Total Functions</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV, $S_2^1$, VPV</td>
<td>$e\mathcal{F}$, $G_1^*$</td>
<td>$P$</td>
<td>[C, B, CN]</td>
</tr>
<tr>
<td>$T_2^1$, $S_2^2$</td>
<td>$G_1$, $G_2^*$</td>
<td>$\leq_{1-1}(PLS)$</td>
<td>[B, KP, KT, BK]</td>
</tr>
<tr>
<td>$T_2^2$, $S_2^3$</td>
<td>$G_2$, $G_3^*$</td>
<td>$\leq_{1-1}(CPLS)$</td>
<td>[B, KP, KT, KST]</td>
</tr>
<tr>
<td>$T_i^i$, $S_i^{i+1}$</td>
<td>$G_i$, $G_i^{*}$</td>
<td>$\leq_{1-1}(LLI_i)$</td>
<td>[B, KP, KT, KNT]</td>
</tr>
<tr>
<td>PSA, $U_2^1$, $W_1^1$</td>
<td>QBF</td>
<td>$PSPACE$</td>
<td>[D, B, S]</td>
</tr>
<tr>
<td>$V_2^1$</td>
<td>**</td>
<td>$EXPTIME$</td>
<td>[B]</td>
</tr>
</tbody>
</table>

| VNC$^1$ | Frege ($\mathcal{F}$) | ALogTime | [CT, A; CM, CN] |
| VL | $GL^*$ | $L$ | [Z, P, CN] |
| VNL | $GNL^*$ | $NL$ | [CK, P, CN] |

PLS = Polynomial local search [JPY]
CPLS = “Colored” PLS [ST]
LLI = Linear local improvement
Paris-Wilkie translation: is a second kind of translation to propositional logic.

- The Paris-Wilkie translation applies to first-order theories with second-order predicates (free variables, $\alpha$), essentially oracles.
- Propositional variables now represent values of the second order objects $\alpha$.
- In contrast, the Cook translation uses variables for the bits of first-order objects (the function’s inputs).
- Paris-Wilkie translations are most commonly applied to fragments of $\text{I} \Delta_0(\#, \alpha)$. [P, PW, ...].

$\alpha$ denotes an uninterpreted second-order object (a predicate, or oracle),
and $\#$ is the polynomial growth rate function $x \# y = 2^{|x| \cdot |y|}$
Let $T$ be the theory $I\Delta_0$ or $I\Delta_0(\#)$.

**Thm:** [PW] If $T(\alpha)$ proves the pigeonhole principle

$$(\forall x \leq a)(\exists y < a)\alpha(x, y) \rightarrow (\exists x < x' \leq a)(\exists y < a)(\alpha(x, y) \land \alpha(x', y))$$

then $\text{PHP}^{n+1}_n$ has polynomial (quasipolynomial, resp) size $\text{AC}^0$-Frege proofs.

Recall $\text{PHP}^{n+1}_n$:

$$\bigwedge_{i=0}^{n} \bigvee_{j=0}^{n-1} p_{i,j} \rightarrow \bigvee_{i=0}^{n-1} \bigvee_{i'=i+1}^{n} \bigvee_{j=0}^{n-1} (p_{i,j} \land p_{i',j})$$

Propositional variables $p_{i,j}$ correspond to truth values of $\alpha(x, y)$. 
On the other hand, [A,BPI,KPW],

**Thm:** $\text{PHP}^{n+1}_n$ requires exponential size $\text{AC}^0$-Frege proofs.

*Proof idea:* apply a Hastad-style switching lemma, to reduce to a proof in which all formulas are decision trees.

**Corollary:** Neither $I\Delta_0$ nor $I\Delta_0(\#)$ proves the pigeonhole principle.

But, [PWW,MPW], ...

**Thm:** $I\Delta(\#)$ proves the weak pigeonhole principle (replacing “$\exists y < a$” with “$\exists y < a/2$”).

**Corollary:** The propositional weak pigeonhole principle $\text{PHP}^{2n}_n$ has quasipolynomial size $\text{AC}^0$-Frege proofs.
A hierarchy of fragments of $\text{I} \Delta_0(\#)$: [B]

- $T^i_2$ - induction for $\Sigma^b_i$ predicates (the $i$-th level of the polynomial time hierarchy).
- $S^i_2$ - length induction for $\Sigma^b_i$ predicates.
- $S^1_2 \subseteq T^1_2 \preceq_{\forall \Sigma^b_2} S^2_2 \subseteq T^2_2 \preceq_{\forall \Sigma^b_3} S^3_2 \subseteq T^3_2 \preceq_{\forall \Sigma^b_4} \cdots$

**Thm:** [KPT]

- If $T^i_2 = S^{i+1}_2$, then the polynomial time hierarchy collapses.
- In fact, if $T^i_2 \preceq_{\forall \Sigma^b_{i+2}} S^{i+1}_2$, then the polynomial time hierarchy collapses.
- $T^i_2(\alpha) \neq S^{i+1}_2(\alpha)$; i.e., relative to an oracle.
$S_2^1(\alpha) \subseteq T_2^1(\alpha) \preceq_{\forall \Sigma_2^b(\alpha)} S_2^2(\alpha) \subseteq T_2^2(\alpha) \preceq_{\forall \Sigma_3^b(\alpha)} \cdots$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_2^1(\alpha), S_2^2(\alpha)$</td>
<td>**</td>
<td>$\leq_{1-1}(PLS(\alpha))$</td>
</tr>
<tr>
<td>$T_2^2(\alpha), S_2^3(\alpha)$</td>
<td>$\text{res(log)}$</td>
<td>$\leq_{1-1}(CPLS(\alpha))$</td>
</tr>
<tr>
<td>$T_2^i(\alpha), S_2^{i+1}(\alpha)$</td>
<td>$\text{depth } (i - \frac{3}{2})$-Frege</td>
<td>$\leq_{1-1}(LLI_i(\alpha))$</td>
</tr>
</tbody>
</table>

Depth $(n + \frac{1}{2})$-Frege means LK proofs with formulas having at most $n+1$ alternations, the bottom level having only logarithmic fanin. $\text{res(log)} = \text{depth } \frac{1}{2}$-Frege.

Sample application: $T_2^2 \vdash \text{PHP}_n^{2n}$. Hence, the bit-graph weak PHP has $\text{res(log)}$ refutations of quasipolynomial size. Likewise, any sparse instance of the weak PHP. [MPW]
Open problem:

(4) Do the theories $T^i_2(\alpha)$ have distinct (increasing) $\forall \Sigma^b_0(\alpha)$-consequences?
   · Note this would not have any (known) computational complexity implications.

(5) For $i \geq 1$, does depth $i$-Frege quasipolynomially simulate depth $(i+1)$-Frege with respect to refuting sets of clauses?
   · Note that this is the nonuniform version of Question (4).

For (5): Best results to-date are a superpolynomial separation, based on upper and lower bounds for the pigeonhole principle. [IK]

Hastad switching lemma gives exponential separation of expressibility in depth $i$ versus depth $i+1$. (!)
(5) asks: Does this extra expressiveness allow shorter proofs?
It is also interesting to study the $\forall \Sigma^b_1$-consequences of the theories $T_i^2$. These define a subset of the TFNP problems:

**Definition:** [MP, P] A **Total NP Search Problem (TFNP)** is a polynomial time relation $R(x, y)$ so that $R$ is

- **Total:** For all $x$, there exists $y$ s.t. $R(x, y)$,

- **Polynomial growth rate:**
  If $R(x, y)$, then $|y| \leq p(|x|)$ for some polynomial $p$.

- The TFNP problem is:
  Given an input $x$, output a $y$ s.t. $R(x, y)$.

Note the solution $y$ may not be unique!
TFNP classes need to come with a proof of totality, usually either a combinatorial principle or a formal proof.

**Pigeonhole Principle (PPP) [P]**

Input: $x \in \mathbb{N}$ and a purportedly injective $f : [x] \to [x-1]$.
Output: $a, b \in [x]$ s.t. either $f(a) \notin [x-1]$ or $f(a) = f(b)$.

**Parity principle (PPAD) [P]**

Input: A directed graph $G$ with in- and out-degrees $\leq 1$, and a vertex $v$ of total degree 1.
Output: Another vertex $v'$ of total degree 1.

**Polynomial Local Search (PLS) [JPY]**

Input: A directed graph with out-degree $\leq 1$, and a nonnegative cost function which strictly decreases along directed edges
Output: A sink vertex.
Proofs in bounded arithmetic also establish TFNP problems:

**PLS** - same as before
**CPLS** - PLS with a Herbrandized coNP ($\Pi^b_1$) accepting condition.

**RAMSEY**
- Input: an undirected graph on $n$ nodes.
- Output: a clique or co-clique of size $\frac{1}{2} \log n$.

But, now the inputs are coded with a second-order object $\alpha$. The output is a first-order object.

**Thm.** The PLS function is provably total in $T^1_2(\alpha)$, and is many-one complete for the provably total relations of $T^1_2(\alpha)$. [BK]

**Thm.** The same holds for CPLS and $T^2_2(\alpha)$. [KST]

**Thm.** $T^3_2(\alpha)$ proves the totality of RAMSEY. [P]

See also: Game Induction [ST], Local Improvement [KNT, BB], ...
Open problems:

(6) Do the $\forall \Sigma^b_1(\alpha)$ consequences (or, the provably total functions) of $T^i_2$ form a proper hierarchy (for $i = 2, 3, 4, \ldots$)?

(7) Does $T^2_2(\alpha)$ prove the totality of RAMSEY?

The $T^3_2(\alpha)$ proof of RAMSEY is essentially a refinement of the usual inductive combinatorial proof of the Ramsey theorem (via a reduction to the pigeonhole principle). It appears that proving RAMSEY in $T^2_2(\alpha)$ would require a new method proof for Ramsey’s theorem.

See also related results and questions for the theory of approximate counting, APC$^2$. [J,KT]
TFNP problems for stronger theories:

**Consistency search** problem for Frege proofs: [BB]
   Input: A (purported) Frege proof of $\bot$.
   Output: A local error in the proof.

Also introduced as the **Wrong proof** search problem [GP].

**Thm.**

- The Frege Consistency Search problem is provable in $U^1_2(\alpha)$ and many-one complete for its provably total functions. [BB]
- The same holds for extended Frege and $V^1_2(\alpha)$. [K, BB]

Here the input is coded by a second-order object; i.e., algorithms have *oracle* access to the Frege “proof” and seek a local error.

The “standard” TFNP problems are all included in the Consistency Search/Wrong Proof search classes for all these theories. [BB, GP]
Finis
Finis

Thank you!
Tseitin, 1968

The question of the minimum complexity of derivation of a given formula in classical propositional calculus is considered in this article and it is proved that estimates of complexity may vary considerably among the various forms of propositional calculus.

Initiating the modern study of proof complexity.
The idea of a refutation-algorithm, [...], is not new. In essence, it goes back to Herbrand, and [is] based on the idea of generating a sequence of quantifier-free lines, and then testing the conjunction of the first \( n \) lines for consistency as \( n = 1, 2, 3, \ldots \) [...]. However, the crucial difficulty, to which little attention appears to have been given in this connection, is that of finding a feasible technique for testing the conjunction of the first \( n \) lines for consistency when \( n \) is large.

Anticipating the hardness of satisfiability for SAT solvers.

Many formal theories have some kind of translation to propositional logic. (!)