Provably Total NP Search Problems

Arnold Beckmann

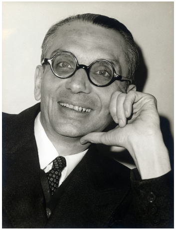
Department of Computer Science
Faculty of Science and Engineering
Swansea University
Wales, UK

World Logic Day 2026
Swansea, 14 January 2026

The Heritage

G. Peano (1858-1932)

A. Turing (1912-1954)



K. Gödel (1906-1978)

G. Gentzen (1909-1945)

Peano Arithmetic (PA)

- domain: $\mathbb{N}$
- language: 0, 1, $+$, $\cdot$, $\leqslant$
- axioms: defining equations

  schema of full induction for all formulas:

$$A(0) \ \wedge \ \forall x(A(x) \to A(x+1)) \ \to \ \forall x A(x)$$

Gödel's 2nd Incompleteness Theorem: $\mathsf{PA} \nvdash \mathsf{Con}_{\mathsf{PA}}$

Gentzen's Consistency Proof: $\mathsf{PA} + \mathrm{TI}(\epsilon_0) \vdash \mathsf{Con}_{\mathsf{PA}}$

Fragments of PA

- Laurie Kriby and Jeff Paris end of 70's:
  $I\Sigma_n$, that is, induction restricted to Σ_n formulas

$$\exists x_1 \forall x_2 \ldots Q x_n \underbrace{\varphi(x_1, \ldots, x_n)}_{\text{bounded quantifiers}}$$

- Charles Parsons beginning of 70's:
  provable recursive functions of $I\Sigma_1$ = *prim. rec. functions.*

- Wainer, Ono & Kadota 70's & '80:
  provable recursive functions of PA = $<\epsilon_0$*-rec. functions.*

Bounded Arithmetic

Cook 1975:
equational theory PV

Buss 1985:
aligned to PA

Zambella 1997, Cook-Nguyen 2010:
language with sorts for strings and indices

The Language of Bounded Arithmetic

- similar to PA
- domain: $\mathbb{N}$
- language: 0, 1, $+$, $\cdot$, $\leqslant$ plus $|.|$, $\#$, ...

$$|x| = \text{binary length of } x$$

$$x\#y = 2^{|x|\cdot|y|} \text{ polynomial growth rate}$$

- bounded formulas:

Σ^b_1 : $\exists x_1 \leqslant s_1 \ \forall y \leqslant |t| \ A(x_1, y)$ — $\mathsf{NP} = \Sigma^p_1$

Σ^b_2 : $\exists x_1 \leqslant s_1 \ \forall x_2 \leqslant s_2 \ \exists y \leqslant |t| \ A(x_1, x_2, y)$ — $\mathsf{NP}^{\mathsf{NP}} = \Sigma^p_2$

$\vdots$

s_1, s_2, t terms, A quantifier-free

Theories of Bounded Arithmetic

BASIC = set of open formulas defining non-logical symbols

Induction:

Σ^b_i-Ind : $\varphi(0) \wedge \forall x(\varphi(x) \to \varphi(x+1)) \to \forall x \varphi(x)$

Σ^b_i-LInd : $\varphi(0) \wedge \forall x(\varphi(x) \to \varphi(x+1)) \to \forall x \varphi(|x|)$

where $\varphi \in \Sigma^b_i$

Theories:

$$\begin{array}{lcl} S^1_2 & = & \text{BASIC} + \Sigma^b_1\text{-LInd} \\ S^2_2 & = & \text{BASIC} + \Sigma^b_2\text{-LInd} \\ & \vdots & \\ S_2 & = & \bigcup_i S^i_2 \end{array} \qquad \begin{array}{lcl} T^1_2 & = & \text{BASIC} + \Sigma^b_1\text{-Ind} \\ T^2_2 & = & \text{BASIC} + \Sigma^b_2\text{-Ind} \\ & \vdots & \\ T_2 & = & \bigcup_i T^i_2 \end{array}$$

Definable Functions

Definition

f Σ_1^b-definable function in S_2^1 iff exists Σ_1^b-formula φ s.t.

- φ defines graph of f over $\mathbb{N}$, i.e. $f(x) = y \iff \mathbb{N} \models \varphi(x, y)$
- $S_2^1 \vdash \forall x\ \exists y \leqslant t(x)\ \varphi(x, y)$ for some term t
- $S_2^1 \vdash \forall x, y, y'\ (\varphi(x, y) \wedge \varphi(x, y') \rightarrow y = y')$

Theorem (Buss '85)

Σ_1^b-*definable functions in* S_2^1 = *p-time functions* FP

Other Characterisations

Theories	Induction	Graph Definability	Computational Complexity
S^1_2	Σ^b_1-LInd	Σ^b_1	FP
S^2_2	Σ^b_2-LInd	Σ^b_2	FP^{NP}
S^{k+1}_2	Σ^b_{k+1}-LInd	Σ^b_{k+1}	$FP^{\Sigma^p_k}$
U^1_2	$\Sigma^{1,b}_1$-LInd	$\Sigma^{1,b}_1$	FPSPACE
V^1_2	$\Sigma^{1,b}_1$-Ind	$\Sigma^{1,b}_1$	FEXPTIME

Results

Theorem (Buss'85, Buss'90)

$$S^1_2 \subseteq T^1_2 \preceq_{\forall\Sigma^b_2} S^2_2 \subseteq T^2_2 \preceq_{\forall\Sigma^b_3} S^3_2 \ \ldots$$

Main Open Problem

Is the hierarchy of theories strict?

Further Results

Independence Results [Krajíček, Pudlák, Takeuti '91, Krajíček'93, Jeřábek '09]

Separation of polynomial time hierarchy PH implies separation of BA theories.

Theorem [Buss '95, Zambella '96, Jeřábek '09]

Collapse of BA theories is equivalent to collapse of PH provable in BA

Propositional Proof Complexity [Cook, Reckhow '79]

Deep connections to propositional proof complexity.

Total NP Search Problems (TFNP)

Definition (Meggido-Papadimitriou '91; Papadimitriou '94)

A *Total NP Search Problem* (TFNP, for *total functional NP*) is a polynomial time computable (ptime), binary relation R, such that

- R is *honest*, i.e. polynomially bounded:
  If $R(x, y)$ then $|y| \leqslant p(|x|)$ for some polynomial p.
- R is *total*: For all x, there exists y s.t. $R(x, y)$.

The *search task* is:
Given input x, find a y s.t. $R(x, y)$.

TFNP intermediate between P and NP

Theorem

If $\mathsf{TFNP} \subseteq \mathrm{FP}$*, then* $\mathsf{NP} \cap \mathsf{coNP} = \mathsf{P}$.

Theorem

If $\mathsf{P} = \mathsf{NP}$*, then* $\mathsf{TFNP} \subseteq \mathrm{FP}$.

Search Complexity Classes in TFNP

Papadimitriou et.al. identified several natural sub-classes of TFNP.

defining principles: combinatorial lemma guaranteeing totality

- PPA *every graph has an even number of odd-degree nodes*
- PPP *there is no injective map from* [x] *to* [x−1]
- PLS *every directed acyclic graph has a sink*
- ...

In this talk: totality guaranteed by mathematical theories like Bounded Arithmetic or Peano Arithmetic.

Provable Total NP Search Problems

Let TFNP(T) be *set of provably total* NP *search problems in* T.

Consider

$$\begin{aligned} \mathrm{FP} = \mathsf{TFNP}(\mathrm{S}^1_2) &\subseteq \mathsf{TFNP}(\mathrm{T}^1_2) \subseteq \mathsf{TFNP}(\mathrm{T}^2_2) \subseteq \cdots \subseteq \mathsf{TFNP}(\mathrm{BA}) \\ &\subseteq \mathsf{TFNP}(\mathrm{I}\Sigma_1) \subseteq \cdots \subseteq \mathsf{TFNP}(\mathrm{PA}) \subseteq \mathsf{TFNP}(\mathrm{ZFC}) \end{aligned}$$

Is this hierarchy strict? Anywhere? (Implies $\mathsf{P} \neq \mathsf{NP}$!)

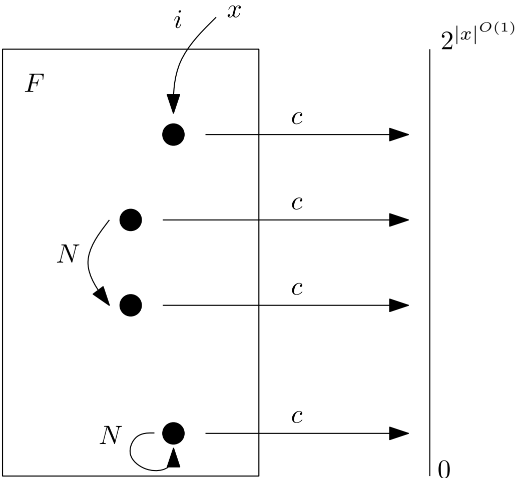
Π^b_k-Polynomial Local Search

Π^b_k-Polynomial Local Search

On input x:
F set of feasible solutions with polynomial bound d
i initial value
N neighbourhood function
c cost function

Search task: Find $s \in F(x)$ with $N(x, s) = s$.

N, i, c ptime, $F \in \Pi^b_k$



A Π^p_k-PLS *problem* $(k > 0)$ on instance x consists of,

- set of feasible solutions $F(x)$,
- neighbourhood function $N(x,s)$,
- cost function $c(x,s)$,
- initial feasible solution $i(x)$,

such that N, c, i polynomial time, F in Π^p_k, and

$$(\forall x,s)(s \in F(x) \to |s| \leqslant d(|x|)) \tag{1}$$

$$(\forall x)(i(x) \in F(x)) \tag{2}$$

$$(\forall x,s)(s \in F(x) \to N(x,s) \in F(x)) \tag{3}$$

$$(\forall x,s)(s \in F(x) \wedge N(x,s) \neq s \to c(x,N(x,s)) < c(x,s)) \tag{4}$$

Search task: find some s with $N(x,s) = s$.

Remark: Π^p_0-PLS, i.e. F ptime, is the same as PLS.

Theorem (Buss, Krajíček'95)

The provably total NP *search problems in* T^1_2 *are exactly characterised by Polynomial Local Search (PLS)*

Proposition

Let $L = (F, N, c, i)$ *be a* Π^b_k*-PLS problem. Then* $(\forall x)(\exists s)(N(x,s) = s)$ *is a provably total* NP *search problem in* T^{k+1}_2.

Proof.

Idea: Consider $\{d \colon (\exists s)(c(x,s) = d \wedge s \in F(x)\}$ $(\in \Sigma^b_{k+1})$.
Provable in T^{k+1}_2, we can determine minimum of this set. Choose some $s \in F(x)$ of minimal cost, this will satisfies $N(x,s) = s$. □

Theorem (B., Buss '09/'10)

Let $0 < k$. *The provably total* NP *search problems in* T^{k+1}_2 *are exactly characterised by* Π^b_k*-PLS problems.*

Other characterisations

CPLS

a version PLS similar to Π^b_1-PLS

Theorem (Kołodziejczyk, Skelly, Thapen '07)

$\mathsf{TFNP}(T^2_2)$ *characterised by CPLS*

k-turn Game Induction Principle, GI_k

k-turn two player games based on induction

Theorem (Skelly, Thapen '11)

$\mathsf{TFNP}(T^k_2)$ *characterised by* GI_k

Local Improvement Principles

Definition (k-round Local Improvement Principle LI_k)

Labels on a directed acyclic graph on $[x]$ can be updated in a consistent, well-founded way for k rounds.

LI (no subscript) allows $k = x$ (exponentially) many rounds

LLI – graph is a line

RLI – graph is a rectangle

Theory	Many-One Complete	
T^k_2	LI_k	[KNT'11]
V^1_2	LI	[KNT'11]
V^1_2	$\mathsf{LI}_{\log}$, LI with $O(\log n)$ may rounds	[BB'14]
U^1_2	LLI, Linear LI	[BB'14]
U^1_2	$\mathsf{LLI}_{\log}$	[KNT'11]
V^1_2	RLI, Rectangular LI	[KNT'11]
V^1_2	$\mathsf{RLI}_{\log}$	[BB'14]
U^1_2	RLI_1	[BB'14]

Provably Total NP Search Problems of Peano Arithmetic

$\prec$-bounded Local Search

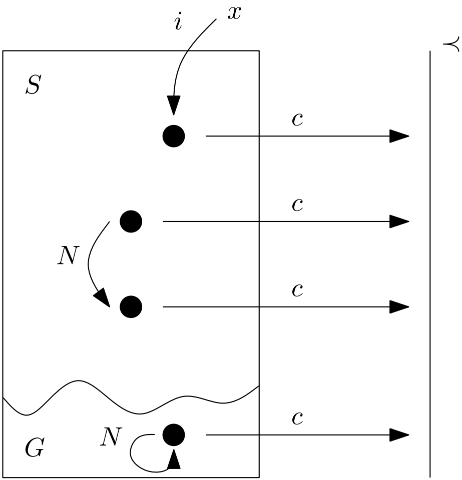
$\prec$ a ptime well-ordering.

S set of possible solutions
S need not be polynomially bounded!
i initial value
N neighbourhood function
c cost function

$$N(x,s) = s \vee c(N(x,s)) \prec c(s)$$

$$N(x,s) = s \;\Rightarrow\; |s| \leqslant d(|x|)$$

Search task:
Find s with $N(x,s) = s$.



A *$\prec$-bounded local search ($\prec$-bls) problem* on input x consists of a set $S(x)$ of possible solutions, a polynomial bound d, a neighbourhood function $\lambda s.N(x,s)\colon S(x) \to S(x)$, a cost function $\lambda s.c(x,s)\colon S(x) \to \mathbb{N}$, a function computing an initial solution $i(x)$,
such that $\prec, S, N, c, i$ are ptime, and

$$\prec \text{ is a total order} \tag{5}$$

$$(\forall x)(i(x) \in S(x)) \tag{6}$$

$$(\forall x, s)(s \in S(x) \to N(x,s) \in S(x)) \tag{7}$$

$$(\forall x, s)(N(x,s) = s \ \lor \ c(x, N(x,s)) \prec c(x,s)) \tag{8}$$

$$(\forall x, s)(N(x,s) = s \to |s| \leqslant d(|x|)) \tag{9}$$

Search task: find some s with $N(x,s) = s$.

Properties of $\prec$-bls Problems

Fact

Any $\prec$-bls problem defines a total NP search problem.

Remark

PLS = $<$-bls with polynomially bounded set of possible solutions.

Definition (Formalised $\prec$-bls Problems)

A $\prec$-bls problem is *formalised* provided the predicates S and $\prec$ are given by Δ^b_1-formulas, and the functions i, N, c are Σ^b_1-definable, such that the $\prec$-bls conditions are provable in S^1_2.

Theorem (B.'09)

The provably total NP search problems in PA are characterised by formalised α-bls problems, for $\alpha \prec_{\epsilon_0} \epsilon_0$.

Proof Idea

Notation for cut-elimination based on Mintz' continuous cut-elimination

Grigori E. Mints. Finite investigations of transfinite derivations. *Journal of Soviet Mathematics*, 10:548–596, 1978.

Wilfried Buchholz. Notation systems for infinitary derivations. *Archive for Math. Logic*, 30:277–296, 1991.

Klaus Aehlig and AB. On the computational complexity of cut-reduction. *APAL*, 161: 711–736, 2010.

A Notation System for (infinitary) propositional logic is a set $\mathcal{D}$ with functions

- for last inference, derived formula, and cut-rank
- $\mathtt{d}[\mathrm{j}]$: a *notation* of jth sub-derivation of (derivation denoted by) $\mathtt{d}$
- $\mathrm{o}(\mathtt{d})$: height of derivation tree (denoted by) $\mathtt{d}$.

AB, S.R.Buss, and C.Pollett. Ordinal Notations and Well-Orderings in Bounded Arithmetic. *APAL*, 120:197–223, 2003.

Further results

TFNP, propositional proof complexity, circuit complexity

- [Göös et.al. '22]
  Relate simulation between propositional proof systems to characterisations of TFNP classes, using this for separations of TFNP classes (relative to an oracle)
- [de Rezende, Göös, Robere '22; Buss, Fleming, Impagliazzo '23]
  TFNP as an organizing principle for connections between propositional proof systems and models of boolean circuits
- [Hubáček, Khaniki, Thapen '24]
  Connections between intersection classes in TFNP and proof properties (feasible disjunction property)

Open Questions / Future Work

- Turn TFNP characterisations into propositional tautologies and show lower bounds in related propositional proof systems for them.
- Separate T^2_2 from T^3_2 with $\forall\Sigma^b_1$ sentences (relativised)
- Analyse TFNP(T) for other theories.
  Conjecture: All theories which admit a good ordinal analysis also admit a characterisation of their provably total NP search problems, similar to PA.

Thanks!