Counting in weak theories

Emil Jeřábek

jerabek@math.cas.cz

http://math.cas.cz/~jerabek/

Institute of Mathematics of the Czech Academy of Sciences, Prague

Logic Colloquium, August 2017, Stockholm

Outline

Finite sets in arithmetic

The language of arithmetic

Arithmetical theories (e.g., Peano arithmetic):

- in theory, the only objects are natural numbers
- in practice, we discuss all kinds of other stuff:
  - sequences, strings, syntactic objects
  - alorithms: recursive functions, Turing machines
  - graphs, finite structures
  - sets

This talk:
we focus on finite sets and their cardinality ("counting")

Finite sets in *PA*

Ways to represent sets in *PA*:

- encode sequences (e.g., Gödel's β-function), represents sets by sequences that enumerate them
- define the graph of exponentiation, use binary expansion

$$u \in x \iff u\text{'th bit of } x \text{ is } 1 \iff \left\lfloor \frac{x}{2^u} \right\rfloor \text{ is odd}$$

- indirectly: bounded definable sets

$$X = \{u < a : \varphi(u, z)\}$$

Choice of representation

Each has its merits

- bounded definable sets: most flexible
- binary expansion: 1–1 representation

In *PA*: all three representations are equivalent

Caveat:

- bounded definable sets $\mapsto$ encoded sets
  $\equiv$ bounded comprehension schema
  $\equiv$ induction

Working with finite sets

What can we do with these sets in *PA*?

- intersection, union, relative complement
- Cartesian product, projection, ...
- in fact: ZF_{fin}

Counting the size:

- if a sequence w is an increasing enumeration of X ("counting function"), put $|X| := \mathrm{lh}(w)$
- *PA* proves $|X \dot\cup Y| = |X| + |Y|$, $|X \times Y| = |X| \cdot |Y|$, ...

Below PA

The full power of PA is not needed

Everything works smoothly in $I\Delta_0 + EXP$ aka EA aka EFA:

- induction for bounded formulas + totality of 2^x
- theory of Kalmár elementary recursive functions
- proves equivalence of representation of finite sets by binary expansion, by sequences, and by bounded $\Delta_0(\exp)$-definable sets ($\approx$ elementary recursive)
- the definition of $|X|$ by counting functions works
- all the expected basic properties hold

Weaker theories?

Without exponentiation, things become interesting

Distinction between

- arbitrary numbers x: LARGE/long/binary
- numbers x s.t. 2^x exists: small/short/unary/lengths

Notation: $\mathrm{Log} = \{x : \exists y\, (2^x = y)\}$

Sequence encoding works, with

- elements: LARGE
- length: small

Sets without exponentiation

Representation matters now!

- sets by binary expansion: small sets of small numbers
- sets as sequences: small sets of LARGE numbers
- bounded definable sets: LARGE sets of LARGE numbers

We are primarily interested in bounded definable sets:

- want simple things like $\{0, \dots, b\}$ to be sets
- most of useful sets are not logarithmically sparse
- NB: we may only allow sets definable by a very restrictive class of formulas

Counting without exponentiation?

Trouble: counting sets by enumeration only works for sets encoded by sequences!

Challenge: Design a method of counting definable sets in theories without exponentiation

Bounded arithmetic

Theories of bounded arithmetic

Bounded formulas: only bounded quantifiers

$$\exists x \le t\, \varphi(x) \iff \exists x\, (x \le t \wedge \varphi(x))$$
$$\forall x \le t\, \varphi(x) \iff \forall x\, (x \le t \to \varphi(x))$$

Þe oldest one [Par'71]: $I\Delta_0$

- induction for Δ_0 formulas = bounded formulas in L_{PA}
- $\Delta_0(\mathbb{N}) = \mathrm{LinH}$ (linear-time hierarchy)
- Parikh's theorem: $I\Delta_0 \vdash \forall x\, \exists y\, \theta(x, y)$, $\theta \in \Delta_0$
  $\implies I\Delta_0 \vdash \forall x\, \exists y \le t(x)\, \theta(x, y)$ for some term t
  - provably total recursive functions
    are bounded by a polynomial

Arithmetic for the polynomial hierarchy

Polynomial time bounds are more interesting than linear time!

- $I\Delta_0 + \Omega_1$: $\forall x\, \exists y\, (y = x^{|x|})$, $|x| = \lceil \log_2(x+1) \rceil$
- Buss's theories: language $\langle 0, 1, +, \cdot, \le, \lfloor x/2 \rfloor, |x|, x \# y \rangle$, where $x \# y = 2^{|x|\,|y|}$
  - T_2 = induction for all bounded (Σ^b_∞) formulas: conservative extension of $I\Delta_0 + \Omega_1$
  - Σ^b_i: i alternating blocks of bounded quantifiers, ignoring sharply bounded quantifiers $\exists x \le |t|$, $\forall x \le |t|$
  - $T^i_2 = \Sigma^b_i$-IND
  - $\Sigma^b_1(\mathbb{N}) = \mathrm{NP}$, $\Sigma^b_i(\mathbb{N}) = \Sigma^{\mathrm{P}}_i$ $(i > 0)$
  - provably total Σ^b_{i+1}-definable functions of T^i_2 are $\mathrm{FP}^{\Sigma^{\mathrm{P}}_i}$

Bigger picture

Proof complexity: (loose) 3-way correspondence between

- theories of arithmetic T
- complexity classes C
- propositional proof systems P

(we mostly ignore P in this talk)

- FC-functions are provably total in T
- T has induction (comprehension, etc.) only for C-predicates
  - "feasible reasoning"

Are basic properties of C provable while reasoning only with C-concepts?

Exact counting in bounded arithmetic

Enumeration by sequences $\implies$ $I\Delta_0$ can count sets up to logarithmic size

[PW'87]: It can also do polylogarithmic size

In $I\Delta_0 + \Omega_1$ and Buss's theories, this makes no difference

We likely can't do better

- Toda's theorem: $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}}$
  if we can count ptime sets by Σ^b_∞ formulas, PH collapses
- Relativization: we cannot count $\Sigma^b_0(\alpha)$-sets of more than polylogarithmic size by $\Sigma^b_\infty(\alpha)$ formulas
  - translate to subexponential constant-depth circuits for Majority

Application of counting

What do we want to count in bounded arithmetic for, anyway?

- formalize randomized algorithms & randomized complexity classes: ZPP, BPP, MA, ...
- formalize probabilistic and counting arguments to prove combinatorial statements
  - Ramsey's theorem: a graph of order n has a clique or independent set of size $\geq \frac{1}{2} \log n$
  - the tournament principle: a tournament with n players has a dominating set of size $\leq \log(n+1)$

Example: BPP

A language L is in BPP if there is a randomized poly-time algorithm $P(w, r)$ such that

$$w \in L \implies \Pr_r[P(w,r) \text{ accepts}] \geq \frac{3}{4}$$
$$w \notin L \implies \Pr_r[P(w,r) \text{ accepts}] \leq \frac{1}{4}$$

Examples:

- Rabin–Miller primality test
- polynomial identity testing

Example: Tournament principle

Theorem: A tournament with n players has a dominating set of size $\leq \log(n+1)$

Proof:

- The expected number of wins of a random player is $n/2$
  $\implies$ fix a player x_0 that wins all but $\leq n/2$ matches
- In the remaining subtournament of size $\leq n/2$,
  fix a player x_1 that wins all but $\leq n/4$ matches
- ...
- We reach zero after $k \leq \log n$ steps. Then $\{x_0, \dots, x_k\}$ is a dominating set

Example: Ramsey's theorem

Theorem: An edge labelling of the complete graph K_n by two colours has a homogeneous set of size $\geq \frac{1}{2} \log n$.

Proof: Let $C\colon \binom{[n]}{2} \to \{0,1\}$ be the labelling:

- Fix a vertex v_0. There is $c_0 \in \{0,1\}$ s.t. $|G_1| \geq n/2$, where $G_1 = \{v : C(\{v_0, v\}) = c_0\}$.
- Fix a vertex $v_1 \in G_1$. There is $c_1 \in \{0,1\}$ s.t. $|G_2| \geq n/4$, where $G_2 = \{v \in G_1 : C(\{v_1, v\}) = c_1\}$.
- ...
- Carry on for $k = \log n$ steps: find vertices $v_0, \dots, v_k$ and $c_0, \dots, c_k \in \{0,1\}$ s.t. $C(\{v_i, v_j\}) = c_i$ for $i < j$
- One colour $c \in \{0,1\}$ occurs $\geq k/2$ times among $c_0, \dots, c_k$. Then $\{v_i : c_i = c\}$ is a homogeneous set.

Lower the expectations

We can't count exactly.

But the examples do not need it:
an approximation will be good enough

Weak pigeonhole principle

PHP

Pigeonhole principle:
a pigeonholes cannot accommodate $b > a$ pigeons
(unless some of them share)

Formalization with relations (multifunctions):

$$
\begin{aligned}
mPHP^b_a(R) = {} & \forall y < b\, \exists x < a\, R(y,x) \\
& \to \exists y < y' < b\, \exists x < a\, \big(R(y,x) \land R(y',x)\big)
\end{aligned}
$$

Variants of PHP

Special cases:

- R is a function: injective PHP

$$iPHP^b_a(g) = \forall y < b\, g(y) < a \to \exists y < y' < b\, g(y) = g(y')$$

- R^{-1} is a function: surjective ("dual") PHP

$$sPHP^b_a(f) = \exists y < b \forall x < a\, f(x) \neq y$$

- both are functions: retraction-pair PHP

$$rPHP^b_a(f, g) = \forall y < b\, g(y) < a \to \exists y < b\, f(g(y)) \neq y$$

Weak PHP

- $mPHP_a^{a+1}$ is an exact counting principle
  not available in bounded arithmetic
- Weak PHP: $b \gg a$, typically: $mPHP_a^{2a}$, $mPHP_a^{a^2}$

Theorem [PWW'88, MPW'02]:

$$T^2_2 \vdash mWPHP(\Sigma^b_1)$$

We can employ variants of WPHP as convenience axioms

For various reasons, the useful variant is *sWPHP* (or *rWPHP*)

Counting with WPHP

Basic idea: witness that $|X| \le a$ by exhibiting a function f such that $f\colon a \twoheadrightarrow X$ (for *sWPHP*) or $f\colon X \hookrightarrow a$ (for *iWPHP*)

Trouble: Where shall we get these functions from?

Ostensibly, WPHP is a passive counting principle:
it says something is impossible, it does not supply any counting functions

Counting with WPHP: examples

Ad hoc counting arguments using WPHP:

- [PWW'88]: T_2 proves the existence of ∞ many primes
  - if there are no primes in $[a, a^{11}]$, conjure up an injection $9a \log a \hookrightarrow 8a \log a$ by manipulating prime factorizations
- [Pud'90]: T_2 proves Ramsey's theorem
  - manipulations of sets in the proof above can be witnessed by explicit counting functions
- Tournament principle? no obvious way how to do it

Can we generalize the method?

Two general setups

Approximate probabilities:

- estimate the size of $X \subseteq 2^n$ within error $2^n/\operatorname{poly}(m)$
  $=$ estimate $\Pr_{x<a}[x \in X]$ within error $1/\operatorname{poly}(m)$
- Δ^b_1 sets can be counted in
  $APC_1 := T^0_2 + sWPHP(\mathrm{FP}) \subseteq T^2_2$
- based on pseudorandom generators

Proper approximate counting:

- estimate the size of $X \subseteq 2^n$ within error $|X|/\operatorname{poly}(m)$
- Σ^b_1 sets can be counted in
  $APC_2 := T^1_2 + sWPHP(\mathrm{FP}^{\mathrm{NP}}) \subseteq T^3_2$
- based on hashing

Approximate probabilities

Size comparison with error

Basic idea: $|X| \le |Y|$ if there is a surjection $Y \twoheadrightarrow X$

Definition:
$X, Y \subseteq 2^n$ definable sets, $\varepsilon \ge 0$

- $X \preceq_\varepsilon Y$ iff there exist $v > 0$ and a circuit

$$C \colon v \times (Y \mathbin{\dot\cup} \varepsilon 2^n) \twoheadrightarrow v \times X$$

- $X \approx_\varepsilon Y$ iff $X \preceq_\varepsilon Y \wedge Y \preceq_\varepsilon X$

It works

Theorem [J'07]: APC_1 proves: If X is defined by a circuit and $\varepsilon^{-1} \in \mathrm{Log}$, there exists s such that $X \approx_\varepsilon s$.

- we can estimate $\Pr_{x<a}[x \in X]$ with error ε by drawing $O(1/\varepsilon)$ independent random samples
  $\implies$ randomized poly-time algorithm
- derandomize using the Nisan–Wigderson pseudorandom generator
- analysis of the generator can be carried out in T^0_2, it provides explicit "counting functions" for X
- *sWPHP* supplies "hard functions" needed by the NW generator

Properties of approximate probabilities

APC_1 also proves:

- $\preceq_\varepsilon$ behaves well wrt $X \cup Y$, $X \smallsetminus Y$, $X \times Y$, ...
- averaging principle
  ("if $\Pr_{x,y}[A(x,y)] \ge p$, there is x s.t. $\Pr_y[A(x,y)] \ge p$")
- Chernoff–Hoeffding inequality
- inclusion-exclusion principle

Applications

Formalization of classes of randomized algorithms
$(\mathrm{TFRP}, \mathrm{BPP}, \mathrm{APP}, \mathrm{MA}, \mathrm{AM}, \dots)$

- straightforward to define using approximate probabilities
- can't expect all of them to be "provably total":
  mostly semantic classes, no known complete problems
- instead, show that the definitions are "well-behaved":
  - amplification of probability of success
  - closure properties (e.g., composition)
  - trading randomness for nonuniformity
  - inclusions between randomized classes and levels of PH

Applications (cont'd)

Formalization of specific randomized algorithms:

- Rabin–Miller primality testing algorithm
- [LC'12]: Edmonds's algorithm (testing existence of perfect matchings)
  Mulmuley–Vazirani–Vazirani (finding perfect matchings)

Another application:
[Pich'14] formalization of the PCP theorem

Approximate counting

Approximate counting: overview

Proper approximate counting:
error relative to size of X, not of the ambient universe

- witness that $|X| \leq s$ using linear hash functions (Sipser's coding lemma)
- equivalent to existence of suitable surjective "counting functions"
- asymmetric: no witness for $|X| \geq s$!
- can count "sparse" sets
  $\implies$ useful for inductive counting arguments

Formalization

For $X \subseteq 2^n$ a definable set, $\varepsilon^{-1} \in \mathrm{Log}$:
$X \precsim_\varepsilon s$ iff there is $\{A_i : i < t\}$, $A_i \in \mathbb{F}_2^{t \times n}$, which isolates a suitable Cartesian power X^d

- $A \in \mathbb{F}_2^{t \times n}$ separates x from $X \subseteq \mathbb{F}_2^n$
  if $Ax \neq Ay$ for every $y \in X \smallsetminus \{x\}$
- $\{A_i : i < k\}$ isolates X
  if every $x \in X$ is separated from X by some A_i

Key result [J'09]:
APC_2 proves, roughly speaking:
If X is Σ^b_1, then up to small error, $X \precsim s$ is equivalent to the existence of a $\mathrm{FP}^{\mathrm{NP}}$ surjection $s^d \twoheadrightarrow X^d$

Properties of approximate counting

APC_2 proves:

- $\precsim_\varepsilon$ agrees with exact counting and $\preceq_\varepsilon$ as much as possible
- $\precsim_\varepsilon$ behaves well wrt $X \cup Y$, $X \times Y$
- averaging principles
- approximate increasing enumeration:
  There are t, s s.t. $s \le t \le \lfloor s(1+\varepsilon) \rfloor$, and non-decreasing $\mathrm{FP}^{\mathrm{NP}}$-retraction pairs

$$t \underset{f'}{\overset{f}{\rightleftarrows}} X \underset{g'}{\overset{g}{\rightleftarrows}} s$$

s.t. f, g are almost 1-to-1, and $\left\lfloor \frac{s}{t} x \right\rfloor \le g(f(x)) \le \left\lceil \frac{s}{t} x \right\rceil$

Applications

- APC_2 can formalize proofs of Ramsey's theorem, tournament principle, ...
- improved collapse of hierarchies:
  if $T^i_2 = S^{i+1}_2$, then $T^i_2 = T_2$ proves $\Sigma^b_{i+1} \subseteq \Delta^b_{i+1}/\,\mathrm{poly}$ and $\Sigma^b_\infty = \mathcal{B}(\Sigma^b_{i+1})$
- [BKT'14] APC_2 proves the ordering principle
  separations between relativized fragments of APC_2
- [BKZ'15] collapse of constant-depth proofs with modular-counting gates

Proofs with modular counting gates

$AC^0[p]$-Frege:

- propositional proof system operating with constant-depth formulas using $\bigwedge$, $\bigvee$, $\neg$, and mod-p connectives
- major open problem: superpolynomial lower bounds?
  - Razborov, Smolensky: exponential circuit complexity lower bound
- [BKZ'15]: quasipolynomial simulation by depth-3 proofs
  - formalize Valiant–Vazirani and Toda's theorem in $APC_2^{\oplus_p \mathrm{P}}$
  - Paris–Wilkie translation of bounded arithmetic to propositional logic

Thank you for attention!

References

- A. Atserias, N. Thapen: The ordering principle in a fragment of approximate counting, ACM Trans. Comp. Logic 15 (2014), paper #2
- S. R. Buss: Bounded arithmetic, Bibliopolis, Naples, 1986
- S. R. Buss, L. A. Kołodziejczyk, N. Thapen: Fragments of approximate counting, JSL 49 (2014), 496–525
- S. R. Buss, L. A. Kołodziejczyk, K. Zdanowski: Collapsing modular counting in bounded arithmetic and constant depth propositional proofs, Trans. AMS 367 (2015), 7517–7563
- S. A. Cook, P. Nguyen: Logical foundations of proof complexity, Cambridge University Press, 2010
- E. Jeřábek: Dual weak pigeonhole principle, Boolean complexity, and derandomization, APAL 129 (2004), 1–37
- ________: Approximate counting in bounded arithmetic, JSL 72 (2007), 959–993
- ________: Approximate counting by hashing in bounded arithmetic, JSL 74 (2009), 829–860
- J. Krajíček: Bounded arithmetic, propositional logic, and complexity theory, Cambridge University Press, 1995

References (cont'd)

- Đ. T. M. Lê and S. A. Cook: Formalizing randomized matching algorithms, LMCS 8 (2012), paper #5
- Đ. T. M. Lê: Bounded arithmetic and formalizing probabilistic proofs, Ph.D. thesis, U. of Toronto, 2014
- A. Maciel, T. Pitassi, A. Woods: A new proof of the weak pigeonhole principle, JCSS 64 (2002), 843–872
- R. Parikh: Existence and feasibility in arithmetic, JSL 36 (1971), 494–508
- J. B. Paris, A. J. Wilkie: Counting delta-zero sets, Fund. Math. 127 (1986), 67–76
- J. B. Paris, A. J. Wilkie, A. R. Woods: Provability of the pigeonhole principle and the existence of infinitely many primes, JSL 53 (1988), 1235–1244
- J. Pich: Logical strength of complexity theory and a formalization of the PCP theorem in bounded arithmetic, LMCS 11 (2015), paper #8
- ________: Complexity theory in feasible mathematics, Ph.D. thesis, Charles U., Prague, 2014
- P. Pudlák: Ramsey's theorem in bounded arithmetic, Proc. CSL '90 (Börger, Büning, Richter, Schönfeld, eds.), LNiCS 533, Springer, 1991, 308–317
- M. Sipser: A complexity theoretic approach to randomness, Proc. 15th STOC (1983), 330–335
- S. Toda: On the computational power of PP and $\oplus\mathrm{P}$, Proc. 30th FOCS (1989), 514–519