Search Problems, Proof Complexity and Second-Order Bounded Arithmetic

Sam Buss

LCC: Logic and Computational Complexity
& ImmermanFest
Vienna, Summer of Logic
July 12, 2014

Second-order Bounded Arithmetic Theories U_2^1 and V_2^1.

U_2^1 and V_2^1 are second-order bounded arithmetic theories for polynomial space (PSPACE) and exponential time ($\mathrm{EXPTIME}$).

[B, 1985]

First-order language for (non-negative) integers:

Symbols: 0, S, $+$, $\cdot$, $\#$, $\lfloor \frac{1}{2}x \rfloor$, $|x|$, $\leq$.

$|x|$ is the length of the binary representation of x.

$x\#y \ := \ 2^{|x|\cdot|y|}$ — gives polynomial growth rate functions.

First-order quantifiers range over integers:

Unbounded quantifiers: $\forall x$, $\exists x$.

Bounded quantifiers: $\forall x \leq t$, $\exists x \leq t$.

Sharply bounded quantifiers: $\forall x \leq |t|$, $\exists x \leq |t|$.

Second-order quantifiers range over sets of integers.

$\forall X$, $\forall Y$. Not explicitly bounded.

Classifications of bounded formulas:

- Σ_i^b, Π_i^b - Formulas with $\leq i$ alternating blocks of bounded first-order quantifiers ignoring sharply bounded quantifiers. No unbounded quantifiers. May contain second-order variables, but no second-order quantifiers.
- $\Sigma_0^{1,b}$ - Formulas with no unbounded quantifiers, and no second-order quantifiers. Equals $\bigcup_i \Sigma_i^b$.
- $\Sigma_i^{1,b}$, $\Pi_i^{1,b}$ - Formulas with i alternating blocks of second order quantifiers, ignoring first-order quantifiers. No unbounded first-order quantifiers.

Normal forms:
By assumption, negations are pushed in to atomic formulas.
Our theories have sufficient comprehension so that, w.l.o.g., sharply bounded quantifiers are pushed inside bounded first-order quantifiers, and bounded first-order quantifiers are pushed inside second-order quantifiers.

Complexity characterizations

- Σ_1^b and Π_1^b formulas express exactly NP and coNP properties.
- Σ_i^b and Π_i^b formulas express exactly properties at the i-th level of the polynomial time hierarchy.
- $\Sigma_1^{1,b}$ formulas express exactly $\mathrm{NEXPTIME}$ properties.

Γ-IND induction axioms

$\varphi(0) \wedge \forall x(\varphi(x) \to \varphi(x+1)) \to \forall x \varphi(x)$; for $\varphi \in \Gamma$.

Γ-PIND induction axioms

$\varphi(0) \wedge \forall x(\varphi(\lfloor \frac{1}{2} x \rfloor) \to \varphi(x)) \to \forall x \varphi(x)$; for $\varphi \in \Gamma$.

The "PIND" axioms are "feasible" forms of induction.

Bounded Arithmetic Theories

"BASIC" - universal axioms giving properties of the function and relation symbols.

$\Sigma_0^{1,b}$-comprehension axioms

$(\forall \vec{x})(\forall \vec{X})(\exists Z)(\forall y \leq t)[y \in Z \leftrightarrow \varphi(y, \vec{x}, \vec{X})]$, for $\varphi \in \Sigma_0^{1,b}$

Definition (S_2^1)

S_2^1 is BASIC + Σ_1^b-PIND.

Definition (U_2^1)

U_2^1 is BASIC + $\Sigma_1^{1,b}$-PIND + $\Sigma_0^{1,b}$-comprehension.

Definition (V_2^1)

V_2^1 is BASIC + $\Sigma_1^{1,b}$-IND + $\Sigma_0^{1,b}$-comprehension.

Definability of functions

Definition

Let Γ be a class of formulas, and T be a theory. Also suppose f is a total (multi)function, $f = f(\vec{a})$ or $f = f(\vec{a}, \vec{A})$. Then f *is* Γ*-definable by* T if, there is some $\phi \in \Gamma$ which defines the graph of f such that

$$T \vdash \forall \vec{x} \exists y \, \phi(\vec{x}, y)$$

or (respectively),

$$T \vdash \forall \vec{x}, \vec{X} \exists y \, \phi(\vec{x}, \vec{X}, y).$$

Theorem ([B'85])

- *The Σ_1^b-definable functions of S_2^1 are precisely the polynomial time functions.*
- *The $\Sigma_1^{1,b}$-definable functions of U_2^1 are the* PSPACE *functions.*
- *The $\Sigma_1^{1,b}$-definable functions of V_2^1 are the* $\mathrm{EXPTIME}$ *functions.*

Remarks

- The inputs $\vec{x}$ are usual inputs. Any second-order inputs $\vec{X}$ are given as oracles.
- For any of these three theories, we can have uniqueness: $\forall \vec{x} \exists! y \phi'(\vec{x}, y)$.
- Characterizing Σ_1^b-definable functions of U_2^1 and V_2^1 is an open problem, and is a main topic of this talk.

Total NP Search Problems (TFNP)

Definition

A *Total NP Search Problem* is given by a polynomial time property $\phi(x, y) = \phi(x, y)$ with inputs x and y such that

$$\forall x \, \exists y \, [|y| \leq p(|x|) \wedge \phi(x, y)].$$

Canonical examples include PLS [JPY'88]; PPAD, PPADS [MP'91,P'94], and many others.

Let T be a true theory, say U_2^1 or V_2^1.
Any Σ_1^b definable function of T is a total NP search problem.

Goal: characterize the provably total NP search problems of U_2^1 and V_2^1.

Definition

Suppose that $(\forall x)(\exists y \leq t)\phi(y, x)$ and $(\forall x)(\exists y \leq s)\psi(y, x)$ specify NP search problems, denoted $y = Q_\phi(x)$ and $y = Q_\psi(x)$. A *many-one reduction* from Q_ϕ to Q_ψ consists of a pair of polynomial time functions g and h such that whenever $y = Q_\psi(g(x))$, we have $h(y, x) = Q_\phi(x)$.
We write $Q_\phi \leq_m Q_\psi$ to denote that there is a many-one reduction from Q_ϕ to Q_ψ.

Definition

A theory proves that $Q_\phi \leq_m Q_\psi$ provided that it proves

$$(\forall x)(\forall y)[y = Q_\psi(g(x)) \rightarrow h(y, x) = Q_\phi(x)]$$

for some explicitly polynomial time functions g and h.

Definition (Local Improvement Principles [KNT'11])

An instance of Local Improvement consists of:

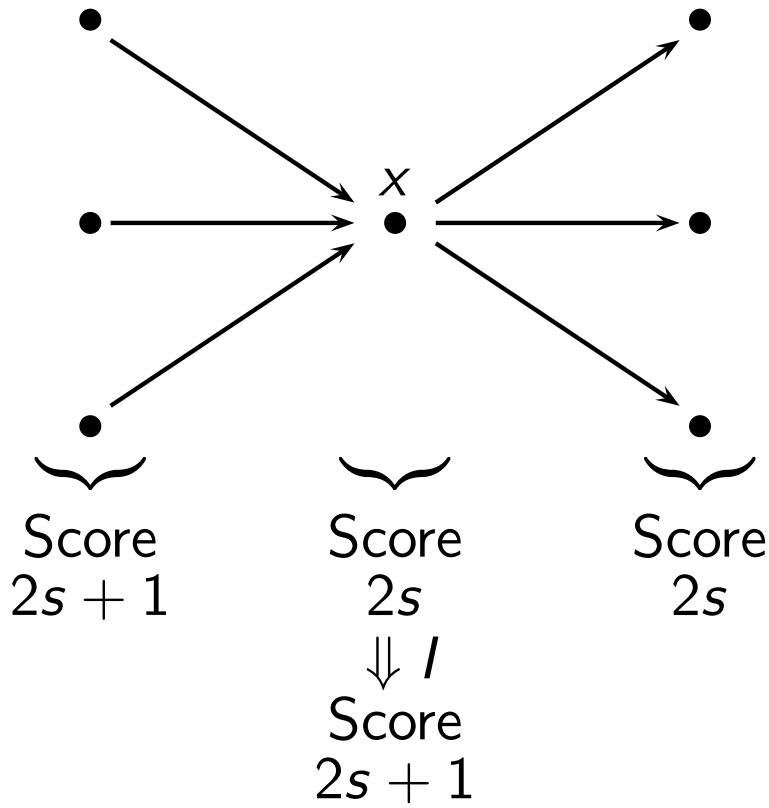
- A constant in- and out-degree dag G with domain $[a] := \{0, 1, 2, \ldots, a-1\}$ with polynomial time function f computing the edges incident on a vertex. The acyclicity is enforced by requiring edges to respect $<$.
- Nodes of G will be assigned a series of "labels". Labels include "score" values.
- A polynomial time initial labeling function E which assigns to each vertex a label with score 0.
- A polynomial time local improvement function I (described on the next slide), which may assign to a vertex with a label of score s a new label of score $s + 1$.
- A polynomial time, predicate wf, which determines if a labeling of a neighborhood of a vertex x is "wellformed".
- There is an upper bound $b > 0$ on labels and an upper bound $c > 0$ on scores for well-formed neighborhoods.

The improvement function I updates labels with incremented scores by "sweeping back-and-forth" across the dag:

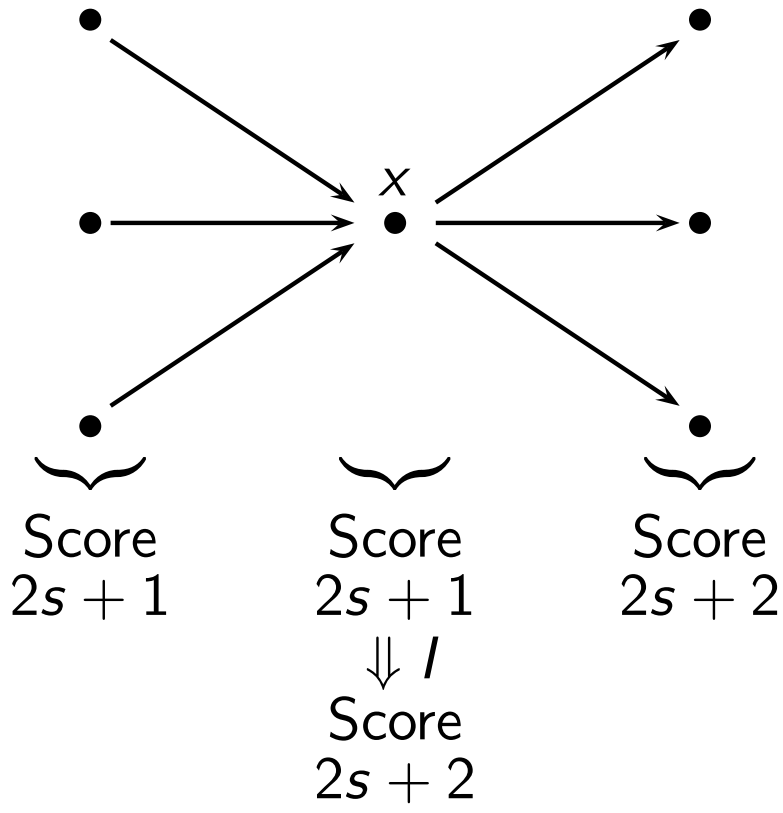
The improvement function I takes as input a wellformed labeling of the neighborhood of a vertex x, and provides a new label for x.

- (Rightward sweep). If s is even, if all predecessors of x have labels with score $s+1$, and if x and all of x's successors have labels with score s; then I provides a new label for x with score $s+1$.
- (Leftward sweep). If s is odd, if all successors of x have labels with score $s+1$, and if x and all of x's predecessors have labels with score s; then I provides a new label for x with score $s+1$.
- In other cases, the I function is undefined.
- A labeling is *extended-wellformed* around x if it is wellformed at x and at each of x's neighbors. The improvement function I preserves this property.

Rightward sweep improvement by I: (even to odd)



Leftward sweep improvement by I: (odd to even)



Definition

The *local improvement principles* state that the above conditions cannot all hold.

Definition

A *solution* to an local improvement principle instance is one of:

- A vertex where G is not a dag respecting $<$.
- A vertex where E fails to give well-formed labels of score 0.
- A local assignment of labels that violate the properties of I or cause I to output a label $> b$ or a score $> c$.

An instance of local improvement uses a parameter a of length $|a| = n$.

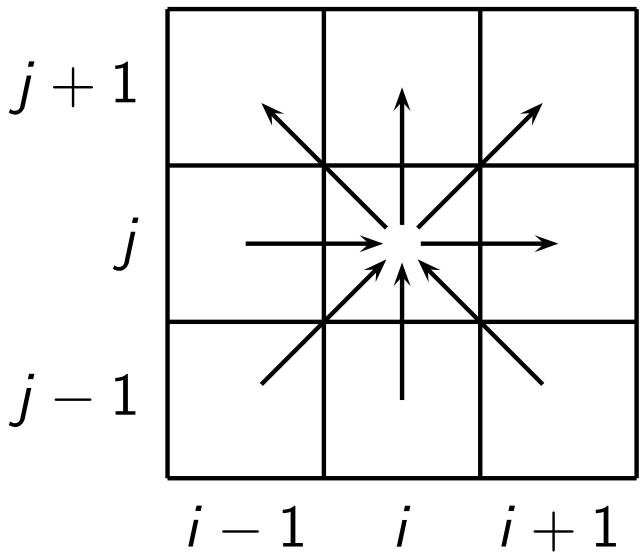
The bound b on labels is always a (wlog).

The bound c on scores may be a (unrestricted), or $|a|$ ("log"), or k (constant $k \geq 1$).

Definition

- LI, LI_{log}, and LI_k are the local improvement principles for arbitrary graphs, with the indicated value for the bound c on the number of rounds (scores).
- LLI, LLI_{log}, LLI_k are the same where the graph G is a dag which is just a line.
- RLI, RLI_{log}, RLI_k are the same where the graph G is a dag in which vertices are rectangularly arranged...

Edges for rectangular dag for RLI, RLI_{log}, and RLI_k:



Theorem ([KNT'11])

RLI *and* LI *are many-one complete for the provably total NP search functions of* V_2^1.

Theorem ([KNT'11])

LLI_{log} *is many-one complete for the provably total NP search functions of* U_2^1.

Remark: as part of being many-one complete problems, LI and LLI_{log} are Σ_1^b-definable total NP search problems of V_2^1 and U_2^1 (respectively).

Theorem ([Beckmann-B'14])

LI_1 *and* RLI_{log} *are many-one complete for the provably total NP search functions of* V_2^1.

Theorem ([Beckmann-B'14])

RLI_1 *and* LLI *are many-one complete for the provably total NP search functions of* U_2^1.

Corollary

1. LLI, LLI_{log} *and* RLI_1 *are equivalent and many-one complete for* U_2^1.
2. LI, LI_{log}, LI_1, RLI, *and* RLI_{log} *are equivalent, and are many-one complete for* V_2^1.

So for $\mathrm{LI}_{...}$ and $\mathrm{LLI}_{...}$, the geometry of the dag is more important than the number of rounds.

Open: What is status for RLI_k for constant $k > 1$?

Proof sketch for $U_2^1 \vdash \mathrm{LLI}$:

First a couple of useful theorems:

"$\Delta_1^{1,b}$" means provably equivalent to a $\Sigma_1^{1,b}$-formula and a $\Pi_1^{1,b}$-formula.

Theorem ([B'85])

- *The $\Delta_1^{1,b}$-predicates of U_2^1 are precisely the PSPACE predicates.*
- *U_2^1 proves $\Delta_1^{1,b}$-IND and $\Delta_1^{1,b}$-MIN.*

Theorem ([Beckmann-B'14])

U_2^1 can formalize Savitch's theorem that $\mathrm{NPSPACE} = \mathrm{PSPACE}$. Thus U_2^1 has induction (IND) and minimization (MIN) for $\mathrm{NPSPACE}$ properties.

Suppose $(G, E, I, b, c, a, \ldots)$ is an instance of LLI.
We want to argue in U_2^1 that some solution exists.

The obvious procedure of applying the initialization E and the improvement function I by sweeping leftward and rightward a times is guaranteed to find a solution.
Unfortunately, this uses exponential space to remember the current labels, and thus is not definable in U_2^1.
(The graph G is implicitly defined on a nodes, and is exponentially large.)

Modify this procedure to be in $\mathrm{NPSPACE}$ by letting it forget scanned labels, and then nondeterministically guessing them as needed on the next scan.

The NSPACE procedure knows labels for vertices $i-2$, $i-1$, i, $i+1$, $i+2$ which are extended well-formed. It uses the I function to update the label (and score) on i, and then advances one vertex left- or rightward.

Any run of the NPSPACE procedure will end with one of:

1. A place where the improvement function I (or, E) fails to give extended well-formed values,
2. A score value $> c$, or
3. A guessed value for a previously generated label fails the needed wellformedness property.

If 1. or 2. occur, they give a solution to the LLI problem.

Consider a longest run of the NPSPACE procedure. It reaches a scan where labels with score $s{+}1$ are being. Say, s is odd and the scan is leftward. For i a vertex in G, consider the scans that reach vertex i when setting scores of $s{+}1$: if there is such a scan which guesses the correct (=previously set) labels of $i{-}1$, i, $i{+}1$, then we call i "good". Take the minimum good value i. It can be shown that any scan which reaches vertex i while setting scores $s{+}1$ halts by finding a place where I fails the needed well-formedness condition; that is, it gives a failure of type 2.

This can be formalized in U_2^1, and $U_2^1 \vdash \mathrm{LLI}$ is proved. $\square$

Propositional proof complexity

Recall: The Cook [C'75] & Paris-Wilkie [PW'87] translations of second-order bounded arithmetic proofs into propositional logic, give a method of translating a valid $\Sigma_0^{1,b}$-formula $\phi(a, X)$ into a family $[\![\phi]\!]$ of tautologies involving propositional variables x_i denoting the value of $X(i)$.

The $[\![\phi]\!]$ translation turns atomic formulas into constants or literals x_i or $\overline{x}_i$. Boolean connectives translate to themselves. Bounded quantifiers translate to large conjunctions or disjunctions.

Theorem

If ϕ is $\Sigma_0^{1,b}$ and $V_2^1 \vdash \phi$, then the tautologies $[\![\phi]\!]$ have quasi-polynomial size extended Frege proofs.

Proof: By the RSUV isomorphism, the conservativity of S_2^1 over PV, and the fact that PV theorems translate to poly size eF proofs. □

Theorem

If ϕ is $\Sigma_0^{1,b}$ and $U_2^1 \vdash \phi$, then the tautologies $[\![\phi]\!]$ have quasi-polynomial size Frege proofs.

Proof: By a similar argument. □

Corollary

The tautologies $[\![\mathrm{LLI}_1]\!]$ have quasi-polynomial size Frege proofs.

Remark: This is surprising, since it appears to need to reason about (concepts equivalent to) the P-complete circuit value problem. □

Corollary

If V_2^1 is conservative over U_2^1 w.r.t. $\Sigma_1^{1,b}$-formulas, then extended Frege proofs can be quasi-polynomially simulated by Frege proofs.

Proof: Since V_2^1 proves the consistency of eF proofs coded by second-objects. □

It is unlikely that V_2^1 is conservative over U_2^1; however, recent developments include:

- Cook-Reckhow's eF proof of the PHP can be quasipolynomially simulated by Frege proofs. [B'??]
- The $AB = I \Rightarrow BA = I$ tautology has quasipolynomial size Frege proofs [Hrubes-Tzameret'12].
- Frankl's Theorem has quasipolynomial size Frege proofs. [Aisenberg-Bonet-B]

We thus have almost no examples of possible tautologies which are known to have short extended Frege proofs, and which plausibly require exponential Frege proofs apart from partial consistency statements.

The various LI and LLI principles, plus RLI_1 and RLI_{log} and RLI are examples of statements which are essentially partial consistency statements.

Question

Can the RLI_k principles, for $k > 2$, give tautologies which (super-quasi-polynomially) separate extended Frege and Frege proofs?

One way to refute this: show $U^1_2 \vdash \mathrm{RLI}_k$ for $k \geq 2$.

Thank you!