Translating Bounded Arithmetic Provability to Frege and Extended Frege Proofs

Paul Beame
University of Washington
Seattle, WA 98195

Two different translations:

- Provability of certain formulas in S_2^1 translates to provability in Extended Frege ($e\mathcal{F}$).

- Provability of formulas in $S_2(R)$ translates to provability in Frege systems using bounded-depth formulas.

Translation of S^1_2 Proofs

We will specify a translation mapping predicate formula $(\forall\vec{x})A(\vec{x})$ where A is bounded to a sequence of propositional formulas $[\![A]\!]_n$, $n \geq 0$ such that:

Thm: (Cook, 1975; Dowd 1985; Buss 1988) If $A \in \Pi^b_2$ and $S^1_2 \vdash (\forall\vec{x})A(\vec{x})$ then there are $e\mathcal{F}$-"proofs" of $[\![A]\!]_n$ of size polynomial in n. Furthermore these $e\mathcal{F}$-"proofs" may be found in polynomial time.

Bounded-depth translation of $S_2(R)$

Let R be a k-place relation symbol.

We'll also specify a translation mapping predicate formula $(\forall x)A(x)$ where A is a bounded arithmetic formula including relation symbol R to a sequence of constant-depth propositional formulas $(A)_n$, $n \geq 0$ and show:

Thm: (Paris, Wilkie 1985) If $I\Delta_0(R) \vdash (\forall x)A(x)$ then there are constant depth Frege proofs of $(A)_n$ of size polynomial in n.
If $I\Delta_0(R)+\Omega_1$ or $S_2(R) \vdash (\forall x)A(x)$ then there are quasi-polynomial ($2^{\log^{O(1)} n}$) size, constant depth Frege proofs of $(A)_n$.

Similar translations will apply for $S_2(R, f)$ and $I\Delta_0(R, f)$ where f is a new k-ary function symbol.

Terms in Bounded Arithmetic formulas

We will show: The lengths of the terms in any $I\Delta_0$ formula $A(\vec{x})$ grow only *linearly* in the lengths of the inputs x_i.

The lengths of the terms in any S_2 formula $A(\vec{x})$ grow only *polynomially* in the lengths of the inputs x_i.

Cor: The values of the terms in any $I\Delta_0$ formula $A(\vec{x})$ grow only *polynomially* in the values of the inputs x_i.

The values of the terms in any S_2 formula $A(\vec{x})$ grow only *quasi-polynomially* in the values of the inputs x_i.

Def: Let t be a term of S_2 (or $I\Delta_0$). The bounding function $q_t(n)$ of t is defined inductively by:

(1) $q_0(n) = 1$

(2) $q_y(n) = n$ for any variable y.

(3) $q_{S(t)}(n) = q_t(n) + 1$ where S is the successor function.

(4) $q_{s+t}(n) = q_s(n) + q_t(n)$

(5) $q_{s \cdot t}(n) = q_s(n) + q_t(n)$

(6) $q_{s\#t}(n) = q_s(n) \cdot q_t(n) + 1$

(7) $q_{|t|}(n) = q_{\lfloor \frac{1}{2} t \rfloor}(n) = q_t(n)$

Prop: If $t(y_1, \ldots, y_k)$ is a term and $x_1, \ldots, x_k$ are natural numbers of length $\leq n$, then $|t(\vec{x})| \leq q_t(n)$ (here $|t(\vec{x})|$ denotes the length of the binary representation of the value of $t(\vec{x})$).

Def: Let A be a bounded formula of $S_2(R)$ (or $I\Delta_0(R)$). The bounding function q_A of A is inductively defined by:

(1) $q_{s=t} = q_{s \le t} = q_s + q_t$

(2) $q_{R(t_1,\ldots,t_k)} = q_{t_1} + \cdots + q_{t_k}$

(3) $q_{A \wedge B} = q_{A \vee B} = q_{A \supset B} = q_A + q_B$

(4) $q_{\neg A} = q_A$

(5) $q_{(\exists x \le t)A}(n) = q_t(n) + q_A(n + q_t(n))$
$$= q_{(\forall x \le t)A}(n)$$

Prop: The formula $A(x_1, \ldots, x_k)$ where $|x_i| \le n$, only refers to numbers of length $\le q_A(n)$.

Observe: If $A(\vec{x})$ is an $S_2(R)$ formula then $q_A(n)$ is a polynomial function. If $A(\vec{x})$ is an $I\Delta_0(R)$ formula then $q_A(n)$ is a linear function.

$S_2(R)$ to Bounded-Depth Frege

A Value-Based Translation

Analog of (Furst, Saxe, Sipser 1981) translation of constant-depth circuit lower bounds to obtain oracle separation of $PSPACE^A$ from PH^A.

input variables $\Leftrightarrow$ answers to oracle queries

Idea: Find constant-depth sequence of propositional formulas $(A)_n$ expressing truth of bounded arithetic formula $A(n)$ in $S_2(R)$ as a function of $R(n_1, \ldots, n_k)$ for $|n_1|, \ldots, |n_k| \leq q_A(|n|)$.

Show: Translation of provability from $S_2(R)$ to bounded-depth Frege

(Ajtai 1983) gave formula translation for $I\Delta_0(R)$. (Paris, Wilkie 1985) showed translation of provability

Let $value_A(n) = 2^{q_A(|n|)}$

Given $S_2(R)$ or $I\Delta_0(R)$ formula $A(x)$ define propositional variables $p_{n_1,\ldots,n_k}$ for all $n_1,\ldots,n_k$ such that each $n_i \leq value_A(n)$

Define propositional translations $(A)_n$ inductively as follows:
(1) If $A(x)$ is an atomic formula with relation $=$ or $\leq$, $(A)_n$ is the constant 0 or 1 given by the value of $A(\underline{n})$.
(2) If $A(x)$ is an atomic formula $R(t_1,\ldots,t_k)(x)$ then $(A)_n$ is $p_{t_1(n),\ldots,t_k(n)}$
(3) Boolean connectives are translated as is:

$$\text{e.g. } (B \wedge C)_n = (B)_n \wedge (C)_n$$

(4) If $A(x) = (\forall y \leq t)B(x,y)$,

$$(A)_n = \bigwedge_{i=0}^{t(n)} (B(x,\underline{i}))_n$$

(5) If $A(x) = (\exists y \leq t)B(x,y)$,

$$(A)_n = \bigvee_{i=0}^{t(n)} (B(x,\underline{i}))_n$$

Note: The size of $(A)_n$ is polynomial in $value_A^d(n)$.

Thm: (Paris, Wilkie 1985) If $I\Delta_0(R) \vdash (\forall x)A(x)$ then there are constant depth Frege proofs of $(A)_n$ of size polynomial in n.
If $I\Delta_0(R)+\Omega_1$ or $S_2(R) \vdash (\forall x)A(x)$ then there are quasi-polynomial ($2^{\log^{O(1)} n}$) size, constant depth Frege proofs of $(A)_n$ for $n \geq 0$.

Proof: The construction of the proofs is identical in the two cases. The only difference is the larger bound on $value_A(n)$ in the $S_2(f)$ case.

Given a proof P of $\forall x A(x)$, instantiate x everywhere in P with $\underline{n}$ to obtain proof $P(n)$ of $A(\underline{n})$.

Apply free-cut elimination $\Rightarrow$ all formulas in P are 'subformulas' of $A(\underline{n})$.

Sequent $\Gamma(\underline{n}) \to \Delta(\underline{n})$ in $P(n)$ may still have free variables.

For each sequent without free variables we let $(\Gamma)_n \to (\Delta)_n$ be the propositional translation applying $(\ )_n$ to each of its formulas.

All axioms and inference rules applied to sequents $\Gamma(\underline{n}) \to \Delta(\underline{n})$ without free variables carry over directly to produce virtually identical Frege proofs of $(\Gamma)_n \to (\Delta)_n$.

Toughest case:

$$\frac{\Gamma \to \Delta, B(s)}{s \le t, \Gamma \to \Delta, (\exists y \le t) B(y)}$$

The lower sequent translates to

$$(s \le t)_n, (\Gamma)_n \to (\Delta)_n, \bigvee_{i=0}^{t(n)} (B(\underline{i}))_n$$

If $s(n) > t(n)$, $(s \le t)_n$ is 0 and the sequent is trivially derived.
If $s(n) \le t(n)$, then $\bigvee_{i=0}^{t(n)} (B(\underline{i}))_n$ follows easily from $(B(s))_n$.

Now consider inferences that contain free variables. Since there are no free variables in $A(\underline{n})$, each free variable appearing in $P(n)$ has a unique elimination inference. Convert these elimination inferences inductively starting at the last elimination inference. This inference is one of:

<u>Case (1):</u> Universal elimination.

$$\frac{(a \le t), \Gamma \to \Delta, B(a)}{\Gamma \to \Delta, (\forall y \le t) B(y)}$$

The bottom sequent has no free variables so it is translated as

$$(\Gamma)_n \to (\Delta)_n, \bigwedge_{i=0}^{t(n)} (B(\underline{i}))_n$$

Let $P(n,i)$ be the substitution of $\underline{i}$ for each occurrence of a in the portion of $P(n)$ that proves the top sequent.

The translation of $P(n)$ will include translations of $P(n,i)$ for each $i = 0, \ldots, t(n)$. These produce sequents:

$$1, (\Gamma)_n \rightarrow (\Delta)_n, (B(\underline{i}))_n$$

for $i = 0, \ldots, t(n)$ from which the bottom line follows by an easy Frege proof.

Case (2): Existential elimination is similar.

Case (3): Induction. Translate IND as easily as PIND.

$$\frac{\Gamma, B(a) \rightarrow B(a+1), \Delta}{\Gamma, B(0) \rightarrow B(t), \Delta}$$

Unwind the induction and as in case (1), produce proofs for copies of the top sequent with a subtituted by $\underline{i}$ for each $i = 0, \ldots, t(n) - 1$. Then an easy Frege proof using cut derives the translation of bottom sequent.

Analysis: The proof P has constant size and therefore has a constant number of elimination inferences. For each elimination the number of formulas grows by a factor of at most $value_A(n)$. Therefore its total number of formulas is at most polynomial in $value_A(n)$. Furthermore, each translated formula is at most polynomial size in $value_A(n)$.

For A an $I\Delta_0(R)$ formula $value_A(n)$ is polynomial in n. For A an $S_2(R)$ formula $value_A(n)$ is quasipolynomial, i.e. $2^{\log^{O(1)} n}$.

The translated formulas have depth at most $i + j$ where i is the quantifier depth of the original formula and j the depth of nesting of its Boolean connectives. Since A is a fixed formula this is a constant. □

A Sharper Bound on Depth

Thm: If S_2^i or $T_2^i \vdash \Gamma(x) \rightarrow \Delta(x)$ then there are depth $i+2$ and $2^{\log^{O(1)} n}$ size Frege proofs of $(\Gamma)_n \rightarrow (\Delta)_n$ for $n \geq 0$. Furthermore, each formula in the proof has bottom fan-in at most $\log^{O(1)} n$.

Proof Sketch: Formulas in Σ_i^b and Π_i^b in general have quantifier depth $\geq i$ because of sharply bound quantifiers. The Quantifier Exchange Principle:

$$\begin{aligned}
&(\forall x \leq |s|)(\exists y \leq t)A(x,y) \\
&\quad \leftrightarrow \quad (\exists y \leq (2s+1)\#(4(2t+1)^2)) \\
&\qquad\qquad (\forall x \leq |s|)A(x, \beta(x+1,y)) \wedge \beta(x+1,y) \leq b
\end{aligned}$$

where we include Godel's β function in the language allows us to collapse all but 2 of the sharply bounded quantifiers.

It is easy to translate the actions of the β function since the translation of terms only depended on computing the outputs of the functions rather than on what functions were involved.

The depth of each $(A)_n$ from the Boolean connectives is easily incorporated in the quantifier depth using a DNF/CNF representation. If the depth is as much as $i+2$, the last quantifier must be sharply bounded. Since any term $|t(n)|$ is most $q_A(|n|)$ in value, we obtain bottom fan-in of at most $\log^{O(1)} n$.

Finally, observe that the previous proof worked for Σ_i^b-IND as well as Σ_i^b-PIND so that the result applies to T_2^i as well as S_2^i. $\square$

S^1_2 and Polysize $e\mathcal{F}$-Proofs

A length-based translation:

Bounded arithmetic formula A in S_2
$\Rightarrow$ sequence of propositional formulas

$$[\![A]\!]_n, \qquad n \geq 0.$$

Desired properties:
(1) $[\![A]\!]_n$ is a propositional formula of size $n^{O(1)}$.
(2) $[\![A]\!]_n$ says that $A(\vec{x})$ is true whenever each $|x_i| \leq n$.
(3) $[\![A]\!]_n$ has polynomial size $e\mathcal{F}$-proofs.

Propositional formulas encoding terms

Encode each S_2 term t in binary assuming its inputs are of length n.
Mimic circuit computation of t on its inputs.
For each function symbol f in S_2,

$$0, S, +, \cdot, |\ |, \#,$$

there are simple fan-out 1 circuits computing the bits of f.

More precisely, we distinguish certain variables of t as *input variables*. Fix some $m \geq n$. For any variable a in t, define propositional variables $v^a_{\ell-1}, \ldots, v^a_0$ encoding the value of a as an ℓ-bit integer where $\ell = n$ for input variables and $\ell = m$ for other variables.

For each S_2 term t, we will define a vector of m propositional formulas

$$[\![t]\!]^n_{m,\vec{x}}$$

giving the low order m bits of the value of t when:

- its variables in $\vec{x}$ are assigned n-bit values,
- its other variables are assigned m-bit values,
- all function evaluations are truncated to their lower order m bits.

(If m is bigger than the length of the value of t we expand the encoding of t with a sequence of leading 0's.)

The total size of $[\![t]\!]^n_{m,\vec{x}}$ will be polynomial in m.

Terms:

(1) $[\![0]\!]^n_{m,\vec{x}}$ is a sequence of m false formulas e.g. $p \wedge \neg p$.

(2) If a is a variable in $\vec{x}$, $[\![a]\!]^n_{m,\vec{x}}$ is a sequence of $m-n$ false formulas followed by $v^a_{n-1}, \ldots, v^a_0$.

(3) If a is a variable not in $\vec{x}$, $[\![a]\!]^n_{m,\vec{x}}$ is $v^a_{m-1}, \ldots, v^a_0$.

(4) $[\![t_1 + t_2]\!]^n_{m,\vec{x}}$ is the vector of formulas obtained by substituting $[\![t_1]\!]^n_{m,\vec{x}}$ and $[\![t_2]\!]^n_{m,\vec{x}}$ into the fan-out 1 circuit for m-bit addition.

(5) The other functions are treated similarly.

Atomic Formulas:

Use fan-out 1 circuits (in fact Boolean formulas) for m-bit $=$, $[\![=]\!]_m$, and m-bit $\leq$, $[\![\leq]\!]_m$.

Define $[\![A]\!]^n_{m,\vec{x}}$ for atomic formulas using these circuits by substituting encodings for their input terms as in (4) and (5) above.

Extension to S_2 Formulas

Given bounded formula A, first convert A to prenex form so that all negated formulas are quantifier-free. In this form define $[\![A]\!]^n_{m,\vec{x}}$ by:

(1) $[\![\neg A]\!]^n_{m,\vec{x}} = \neg[\![A]\!]^n_{m,\vec{x}}$

(2) $[\![A \circ B]\!]^n_{m,\vec{x}} = [\![A]\!]^n_{m,\vec{x}} \circ [\![B]\!]^n_{m,\vec{x}}$
where $\circ = \vee, \wedge, \supset$.

(3) $[\![(\forall y \leq |t|)A(y)]\!]^n_{m,\vec{x}} = \bigwedge_{k=0}^{m-1} [\![\neg \underline{k} \leq |t| \vee A(\underline{k})]\!]^n_{m,\vec{x}}$

(4) $[\![(\exists y \leq |t|)A(y)]\!]^n_{m,\vec{x}} = \bigvee_{k=0}^{m-1} [\![\underline{k} \leq |t| \wedge A(\underline{k})]\!]^n_{m,\vec{x}}$

(5) $[\![(\exists y \leq t)A(y)]\!]^n_{m,\vec{x}}$ is

$$[\![(b \leq t) \wedge A(b)]\!]^n_{m,\vec{x}}(\{\varepsilon_i^A/v_i^b\}_{i=0}^{m-1})$$

where t is not of the form $|s|$, b is a new free variable not in $\vec{x}$ nor occuring in $A(y)$, and $\varepsilon_1^A, \ldots, \varepsilon_m^A$ are new propositional variables.

(6) $[\![(\forall y \leq t)A(y)]\!]^n_{m,\vec{x}}$ is

$$[\![(b \leq t) \wedge A(b)]\!]^n_{m,\vec{x}}(\{\mu^A_i/v^b_i\}^{m-1}_{i=0})$$

where t is not of the form $|s|$, b is a new free variable not in $\vec{x}$ nor occuring in $A(y)$, and $\mu^A_1, \ldots, \mu^A_m$ are new propositional variables.

Note: We call $\varepsilon^A_1, \ldots, \varepsilon^A_m$ 'existential' propositional variables and $\mu^A_1, \ldots, \mu^A_m$ 'universal' propositional variables.
In introducing these variables, new versions of the 'existential' variables are defined for each *instance* of A whereas new versions of the 'universal' variables are only defined for different formulas.

Prop: For fixed formula $A(\vec{x}) \in \Pi^b_2$ and $m \geq q_A(n)$, the propositional formula $[\![A]\!]^n_{m,\vec{x}}$ is polynomial size in m. Moreover $[\![A]\!]^n_{m,\vec{x}}$ expresses 'A is true' in the sense that if A is true for every assignment of numbers of length $\leq n$ to $\vec{x}$, then for any asssignment of truth values to the 'universal' variables and to the v^b_i's, there is a truth assignment to the existential variables that makes $[\![A]\!]^n_{m,\vec{x}}$ true.

Thm: If $A(\vec{x}) \in \Pi^b_2$ and $S^1_2 \vdash \forall \vec{x} A(\vec{x})$, then for any polynomial $q(n) \geq q_A(n)$ there is a sequence of extensions E of the existential variables of $[\![A]\!]^n_{q(n),\vec{x}}$ such that $[\![A]\!]^n_{q(n),\vec{x}}$ has $e\mathcal{F}$-proofs with E of size polynomial in n.

Proof: Assume wlog that S_2 formulas are in prenex form. Show by induction on number of inferences that if $\Gamma \rightarrow \Delta$ is provable in S_2^1, then theorem holds for $[\![\neg\Gamma \vee \Delta]\!]$.
By free-cut elimination can assume $\Gamma \subset \Sigma_1^b$ and $\Delta \subset \Pi_2^b$.

Case (1): $B \rightarrow B$ where B is atomic.

$$[\![\neg B \vee \neg B]\!] = \neg[\![B]\!] \vee [\![B]\!]$$

has trivial $e\mathcal{F}$-proof.

Case (2): BASIC and equality axioms. These are simple polysize circuits so $e\mathcal{F}$ gives simple proofs. Can even give Frege proofs (Buss 1987)

Case (3): Structural Rules. The only non-trivial one is contraction, say:

$$\frac{\Gamma \to \Delta, B, B}{\Gamma \to \Delta, B}$$

Each instance of B has different existential variables, say $\vec{\varepsilon}, \vec{\varepsilon}', \vec{\varepsilon}''$. By induction there is a set E of existential extensions s.t. there are polysize $e\mathcal{F}$-proofs with E of

$$[\![\neg\Gamma \vee \Delta \vee B \vee B]\!]$$

Modify these proofs by adding the following extensions to E

$$\varepsilon_i'' \leftrightarrow ([\![B]\!](\vec{\varepsilon}) \wedge \varepsilon_i) \vee (\neg[\![B]\!](\vec{\varepsilon}) \wedge \varepsilon_i')$$

and use these extensions and the inductive hypothesis as part of an $e\mathcal{F}$-proof.

Case (4): Boolean connective rules. These are directly simulated in $e\mathcal{F}$.

Case (5): Cut.

$$\frac{\Gamma \rightarrow \Delta, B \qquad B, \Pi \rightarrow \Lambda}{\Gamma, \Pi \rightarrow \Delta, \Lambda}$$

By free-cut elimination, $B \in \Sigma_1^b$; so B may have existential variables $\vec{\varepsilon}$ and $\neg B$ may only have universal variables. Apply inductive hypothesis to get polysize $e\mathcal{F}$-proofs with extensions E of

$$[\![\neg\Gamma]\!] \vee [\![\Delta]\!] \vee [\![B]\!](\vec{\varepsilon})$$

and polysize $e\mathcal{F}$-proofs with E' of

$$[\![\neg\Pi]\!] \vee [\![\Lambda]\!] \vee [\![\neg B]\!](\vec{\mu}).$$

Substitute ε_i for μ_i in the latter proof and these easily derive polysize $e\mathcal{F}$-proofs with $E \cup E'$. (There are no existential variables in common.)

Case (6): Σ_1^b-PIND inferences. Unwind the induction as a polynomial length series of cut inferences and contractions.

Case (7): Bounded Quantifier rules. For example

$$\frac{\Gamma \longrightarrow B(s), \Delta}{s \leq t, \Gamma \longrightarrow (\exists x \leq t) B(x), \Delta}$$

Let $\vec{\varepsilon}$ be the existential variables for the variable x in $[\![(\exists x \leq s)A]\!]$. By induction hypothesis there are polynomial-size $e\mathcal{F}$-proofs with extensions E of

$$[\![\neg\Gamma \vee \Delta \vee A(t)]\!]$$

Further derive

$$[\![\neg t \leq s \vee \neg\Gamma \vee \Delta \vee (t \leq s \wedge A(t))]\!]$$

Now add extensions $\vec{\varepsilon} \leftrightarrow [\![t]\!]$ to E and applying these extension to some $[\![t]\!]$ derive

$$[\![\neg t \leq s \vee \neg\Gamma \vee \Delta \vee (\exists x \leq s) A(t)]\!]$$

Case (8): Sharply bounded quantifier rules. For example

$$\frac{\Gamma \to B(s), \Delta}{s \leq |t|, \Gamma \to \exists x \leq |t| B(x), \Delta}$$

Use propositional proof by cases for each of the $q_A(n)$ values $\underline{k}$ that s might evaluate to. This will generate extra copies of formulas in Γ and Δ which can be eliminated by contraction.

$\Box$

Related Formulations

Cook's original version of this result was for the equational system PV which has a much richer language with a function symbol for every polynomial-time function.

(PV is conservative over S^1_2 since every poly-time function may be defined in S^1_2.)

This nicely avoids the 'existential variables' while remaining fairly expressive.

Thm: (Cook 1975) If $s = t$ is a theorem of PV then $[\![s = t]\!]_n$ has a polynomial-size $e\mathcal{F}$-proof.

The same applies to formulas of PV^1, which allow Boolean combinations of equations (but still no quantifiers.)

Krajíček and Pudlák extended the formulation to *Quantified Propositional Calculus (QPC)* - a more powerful notion than $e\mathcal{F}$-proofs.

They define a proof system G_i where corresponds to i alternations of propositional quantifiers and show that for a QPC translation $\langle A \rangle_n$:

Thm: (Krajíček, Pudlák 1988) If $S_2^i \vdash \forall \vec{x} A(\vec{x})$ where $A \in \Pi_2^b$ then $[\![A]\!]_n$ is a Π_2^q formula and has polynomial-size proofs in G_i.