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 \( S^1_2 \) 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 $(\forall \bar{x})A(\bar{x})$ 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 $S^1_2 \vdash (\forall \bar{x})A(\bar{x})$ 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.
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 $I\Delta_0(R) \vdash \Omega_1$ 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 $A(\bar{x})$ grow only linearly in the lengths of the inputs $x_i$.

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

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

The values of the terms in any $S_2$ formula $A(\bar{x})$ 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 $|t(\bar{x})| \leq q_t(n)$ (here $|t(\bar{x})|$ denotes the length of the binary representation of the value of $t(\bar{x})$).
**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. \( q_{s\equiv t} = q_{s \leq t} = q_s + q_t \)
2. \( q_{R(t_1, \ldots, t_k)} = q_{t_1} + \cdots + q_{t_k} \)
3. \( q_{A \land B} = q_{A \lor B} = q_{A \lor B} = q_A + q_B \)
4. \( q_{\neg A} = q_A \)
5. \( q_{(\exists x \leq t)A}(n) = q_t(n) + q_A(n + q_t(n)) = q_{(\forall x \leq t)A}(n) \)

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

**Observe:** If \( A(\bar{x}) \) is an \( S_2(R) \) formula then \( q_A(n) \) is a polynomial function. If \( A(\bar{x}) \) 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 arithmetic 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

(Ajtaï 1983) gave formula translation for $I\Delta_0(R)$. (Paris, Wilkie 1985) showed translation of provability
Let \( \text{value}_A(n) = 2^{q_A(|n|)} \)

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

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 \( A(n) \).
(2) If \( A(x) \) is an atomic formula \( R(t_1, \ldots, t_k)(x) \) then \( (A)_n \) is \( p_{t_1(n)}, \ldots, p_{t_k(n)} \)
(3) Boolean connectives are translated as is:
   e.g. \( (B \land C)_n = (B)_n \land (C)_n \)
(4) If \( A(x) = (\forall y \leq t)B(x, y) \),
   \[ (A)_n = \bigwedge_{i=0}^{t(n)} (B(x, i))_n \]
(5) If \( A(x) = (\exists y \leq t)B(x, y) \),
   \[ (A)_n = \bigvee_{i=0}^{t(n)} (B(x, i))_n \]

**Note:** The size of \( (A)_n \) is polynomial in \( \text{value}_A^d(n) \).
**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 $\text{value}_A(n)$ 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 $\Gamma(\underline{n}) \rightarrow \Delta(\underline{n})$ in $P(n)$ may still have free variables.
For each sequent without free variables we let $(\Gamma)_n \rightarrow (\Delta)_n$ be the propositional translation applying $(\ )_n$ to each of its formulas.

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

Toughest case:

$$
\frac{\Gamma \rightarrow \Delta, B(s)}{s \leq t, \Gamma \rightarrow \Delta, (\exists y \leq t) B(y)}
$$

The lower sequent translates to

$$(s \leq t)_n, (\Gamma)_n \rightarrow (\Delta)_n, \bigvee_{i=0}^{t(n)} (B(i))_n$$

If $s(n) > t(n)$, $(s \leq t)_n$ is 0 and the sequent is trivially derived.

If $s(n) \leq t(n)$, then $\bigvee_{i=0}^{t(n)} (B(i))_n$ follows easily from $(B(s))_n$. 
Now consider inferences that contain free variables. Since there are no free variables in $A(n)$, 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:

**Case (1): Universal elimination.**

$$
\frac{(a \leq t), \Gamma \rightarrow \Delta, B(a)}{\Gamma \rightarrow \Delta, (\forall y \leq t)B(y)}
$$

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

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

Let $P(n, i)$ be the substitution of $i$ 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:

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

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.

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

Unwind the induction and as in case (1), produce proofs for copies of the top sequent with $a$ substituted by $i$ 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 $S^i_2$ or $T^i_2 \vdash \Gamma(x) \rightarrow \Delta(x)$ 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 $\Sigma^b_i$ and $\Pi^b_i$ in general have quantifier depth $\geq i$ because of sharply bound quantifiers. The Quantifier Exchange Principle:

$$(\forall x \leq |s|)(\exists y \leq t)A(x,y)$$

$$(\exists y \leq (2s + 1)\#(4(2t + 1)^2))$$

$$(\forall x \leq |s|)A(x,\beta(x + 1,y)) \land \beta(x + 1,y) \leq b$$

where we include Godel’s $\beta$ 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 $\beta$ 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 $\Sigma^b_i$-IND as well as $\Sigma^b_i$-PIND so that the result applies to $T^i_2$ as well as $S^i_2$. □
$S^1_2$ and Polysize $eF$-Proofs

A length-based translation:

Bounded arithmetic formula $A$ in $S_2$ 
⇒ sequence of propositional formulas 
\[ [A]_n, \quad n \geq 0. \]

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

$$[t]_{m,\vec{x}}^n$$

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 $[t]_{m,\vec{x}}^n$ will be polynomial in $m$. 
Terms:
(1) \([0]^n_{m,\vec{x}}\) is a sequence of \(m\) false formulas
e.g. \(p \land \neg p\).

(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 $[A]^n_{m,\bar{x}}$ by:

1. $[-A]^n_{m,\bar{x}} = \neg [A]^n_{m,\bar{x}}$

2. $[A \circ B]^n_{m,\bar{x}} = [A]^n_{m,\bar{x}} \circ [B]^n_{m,\bar{x}}$
   where $\circ = \lor, \land, \supset$.

3. $[\forall y \leq |t|] A(y)^n_{m,\bar{x}} = \bigwedge_{k=0}^{m-1} [-k \leq |t| \lor A(k)]^n_{m,\bar{x}}$

4. $[\exists y \leq |t|] A(y)^n_{m,\bar{x}} = \bigvee_{k=0}^{m-1} [k \leq |t| \land A(k)]^n_{m,\bar{x}}$

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

   $[b \leq t] \land A(b)^n_{m,\bar{x}} (\{\epsilon_i^A/v_i^b\}_{i=0}^{m-1})$

   where $t$ is not of the form $|s|$, $b$ is a new free variable not in $\bar{x}$ nor occurring in $A(y)$, and $\epsilon_1^A, \ldots, \epsilon_m^A$ are new propositional variables.
(6) \[ (\forall y \leq t)A(y) \] is
\[ \exists (b \leq t) \wedge A(b)\{\mu^A_i / v^b_i \}_{i=0}^{m-1} \]
where \( t \) is not of the form \(|s|\), \( b \) is a new free variable not in \( \bar{x} \) nor occurring 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 $A(\bar{x}) \in \Pi^b_2$ and $m \geq q_A(n)$, the propositional formula $[A]^n_{m,\bar{x}}$ is polynomial size in $m$. Moreover $[A]^n_{m,\bar{x}}$ expresses ‘$A$ is true’ in the sense that if $A$ is true for every assignment of numbers of length $\leq n$ to $\bar{x}$, then for any assignment of truth values to the ‘universal’ variables and to the $v_i^b$’s, there is a truth assignment to the existential variables that makes $[A]^n_{m,\bar{x}}$ true.

**Thm:** If $A(\bar{x}) \in \Pi^b_2$ and $S^1_2 \vdash \forall \bar{x} A(\bar{x})$, then for any polynomial $q(n) \geq q_A(n)$ there is a sequence of extensions $E$ of the existential variables of $[A]^n_{q(n),\bar{x}}$ such that $[A]^n_{q(n),\bar{x}}$ 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 $S^1_2$, then theorem holds for $[\neg \Gamma \lor \Delta]$. By free-cut elimination can assume $\Gamma \subset \Sigma^b_1$ and $\Delta \subset \Pi^b_2$.

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

$$[\neg B \lor \neg B] = \neg [B] \lor [B]$$

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:

\[
\frac{\Gamma \rightarrow \Delta, B, B}{\Gamma \rightarrow \Delta, B}
\]

Each instance of \(B\) has different existential variables, say \(\bar{e}, \bar{e}', \bar{e}''\). By induction there is a set \(E\) of existential extensions s.t. there are polysize \(eF\)-proofs with \(E\) of

\[ [\neg \Gamma \lor \Delta \lor B \lor B] \]

Modify these proofs by adding the following extensions to \(E\)

\[ \bar{e}''_i \leftrightarrow ([B](\bar{e}^* \land e_i) \lor (\neg [B](\bar{e}^*) \land e'_i)) \]

and use these extensions and the inductive hypothesis as part of an \(eF\)-proof.

Case (4): Boolean connective rules. These are directly simulated in \(eF\).
Case (5): Cut.

\[
\frac{\Gamma \rightarrow \Delta, B}{\Gamma, \Pi \rightarrow \Delta, \Lambda}
\]

By free-cut elimination, \(B \in \Sigma^b_1\); so \(B\) may have existential variables \(\varepsilon^i\) and \(\neg B\) may only have universal variables. Apply inductive hypothesis to get polysize \(eF\)-proofs with extensions \(E\) of

\[
[-\Gamma] \lor [\Delta] \lor [B](\varepsilon^i)
\]

and polysize \(eF\)-proofs with \(E'\) of

\[
[-\Pi] \lor [\Lambda] \lor [-B](\mu^i).
\]

Substitute \(\varepsilon^i\) for \(\mu^i\) in the latter proof and these easily derive polysize \(eF\)-proofs with \(E \cup E'\). (There are no existential variables in common.)

Case (6): \(\Sigma^b_1\)-PIND inferences. Unwind the induction as a polynomial length series of cut inferences and contractions.
Case (7): Bounded Quantifier rules. For example

\[
\Gamma \rightarrow B(s), \Delta \\
\frac{s \leq t, \Gamma \rightarrow (\exists x \leq t)B(x), \Delta}{\neg } \]

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

\[\neg \Gamma \lor \Delta \lor A(t)\]

Further derive

\[\neg t \leq s \lor \neg \Gamma \lor \Delta \lor (t \leq s \land A(t))\]

Now add extensions \( \bar{e} \leftrightarrow [t] \) to \( E \) and applying these extension to some \([t] \) derive

\[\neg t \leq s \lor \neg \Gamma \lor \Delta \lor (\exists x \leq s)A(t)\]
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 \(k\) that \(s\) might evaluate to. This will generate extra copies of formulas in \(\Gamma\) and \(\Delta\) which can be eliminated by contraction.

\(\square\)
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 $S_2^1$ since every poly-time function may be defined in $S_2^1$.)

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 $eF$-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 $A \in \Pi^b_2$ then $[A]_n$ is a $\Pi^q_2$ formula and has polynomial-size proofs in $G_i$. 