These comments refer to Arnold Beckmann’s paper [Bec05]. That paper introduces the notion of the uniform reduct of a propositional proof system, which consists of a collection of $\Delta_0(\alpha)$ formulas, where $\alpha$ is a unary relation symbol. Here I will define essentially the same thing, but make it a collection of $\Sigma^B_0$ formulas instead. The $\Sigma^B_0$ formulas (called $\Sigma^p_0$ by Zambella) are two-sorted formulas which are the same as bounded formulas of Peano arithmetic, except that they are allowed free “string” variables $X, Y, Z, ...$ which range over finite sets of natural numbers. Terms of the form $|X|$ are allowed, which denote the “length” of the string $X$ (more precisely 1 plus the largest element of $X$, or 0 if $X$ is empty). The atomic formula $X(t)$ means $t$ is a member of $X$.

Each $\Sigma^B_0$ formula $\varphi(X)$ translates into a family $\langle \varphi(X)[n] : n \in \mathbb{N} \rangle$ of propositional formulas (see [Coo05, CN]) in the style of the Paris-Wilkie translation. The difference is that now $X$ has a length $|X|$, and this affects the semantics of $\varphi(X)$ and the resulting translation. For each $n \in \mathbb{N}$ the propositional translation $\varphi(X)[n]$ of $\varphi(X)$ has atoms $p_0^X, \ldots, p_{n-2}^X$ representing the bits of the string $X$, and $\varphi(X)[n]$ is a tautology iff $\varphi(X)$ holds for all strings $X$ of length $n$. If $\varphi(\vec{X})$ has several string variables $\vec{X} = X_1, \ldots, X_k$ then the translation is the family $\varphi(\vec{X})[\vec{n}]$ of formulas, where $n_i$ is intended to be the length of $X_i$.

In terms of $\Sigma^B_0$ formulas, the definition of uniform reduct in [Bec05] becomes

**Definition:** (Beckmann)

$$U_f = \{ \varphi(\vec{X}) \in \Sigma^B_0 : \langle \varphi(\vec{X})[n] : n \in \mathbb{N} \rangle \text{ has polysize } f\text{-proofs} \}$$

Problem 2 in [Bec05] asks (in our terminology) whether there is a proof system $f$ such that $U_f = \text{TRUE}_{\Sigma^B_0}$ (referring to the set of true $\Sigma^B_0$ formulas).

Here we point out that a positive answer to Problem 2 is equivalent to the existence of an optimal proof system.

Let $f^+$ be the system $f$ augmented to allow substitution Frege rules to be applied to tautologies after exhibiting their $f$ proofs.

**Theorem 1:** $U_{f^+} = \text{TRUE}_{\Sigma^B_0}$ iff $f^+$ simulates every proof system.

**Proof:**

$\Leftarrow:$ For each $\Sigma^B_0$ formula $\varphi(\vec{X})$ we can easily define a proof system in which $\langle \varphi(\vec{X})[n] : n \in \mathbb{N} \rangle$ has polysize proofs.

$\Rightarrow:$ Assume $U_{f^+} = \text{TRUE}_{\Sigma^B_0}$ and let $g$ be any proof system. The idea is to formulate the soundness of $g$ as a $\Sigma^B_0$ formula $\text{Sound}_g$ and then show that $\text{EF}$, using the propositional translations of $\text{Sound}_g$ as axioms, simulates $g$. 

1
This is similar to Theorem 14.1.2 in [Kra95], which states that \( EF + \|0 - RFN(g)\| \) p-simulates \( g \), except soundness of \( g \) is now formulated by the formula \( 0 - RFN(g) \), which is not \( \Sigma^B_0 \). (See also [KP89].)

To formulate soundness of \( g \) by a \( \Sigma^B_0 \) formula we use a \( \Sigma^B_0 \) formula \( \text{Eval}(X, Y, Z) \) which asserts that \( X \) is a truth assignment to the atoms of the formula \( Y \), and \( Z \) extends that assignment to the subformulas of \( Y \) (see Definition 9.3.1.4 in [Kra95]). (The string \( Z \) includes parsing information for the formula \( Y \).)

The proof system \( g \) is a polynomial time map taking strings onto the set of tautologies. Let \( \varphi_g(U, Y, W) \) be a \( \Sigma^B_0 \) formula which asserts that \( W \) is a computation showing that 

\[
\varphi_g(U, Y, W) = Y
\]

Then we define

\[
\text{Sound}_g(U, W, X, Y, Z) = \text{Eval}(X, Y, Z) \land \varphi_g(U, Y, W) \supset Z(0)
\]

where we have rigged the formula \( \text{Eval} \) so that \( Z(0) \) is the truth value of the entire formula \( Y \).

If \( g \) is a proof system, then the universal closure of \( \text{Sound}_g \) is true, and hence its propositional translations \( \text{Sound}_g[\vec{n}] \) have polynomial size \( f+ \) proofs.

Now let \( U_0 \) be a string which is a \( g \)-proof of a formula \( A \), so \( g(U_0) = A \). Let \( W_0 \) be a computation showing \( g(U_0) = A \).

Let \( \text{Sound}'_g(X, Z) \) be the result of substituting \( U_0, A, W_0 \) for \( U, Y, W \) in \( \text{Sound}_g \), and simplifying \( \varphi_g(U_0, A, W_0) \) to 1. Thus

\[
\text{Sound}'_g(X, Z) = \text{Eval}(X, A, Z) \supset Z(0)
\]

Then (for suitable \( k, m \)) the translation \( \text{Sound}'_g(X, Z)[k, n] \) can be obtained by a short substitution Frege proof from the tautologies \( \text{Sound}_g[\vec{n}] \). Now we continue this substitution Frege proof by substitutions in \( \text{Soundg'}_g(X, Z)[k, n] \) as follows:

Substitute the atoms \( q_1, ..., q_\ell \) of the formula \( A \) for the corresponding atoms \( p^X_0, \cdots, p^X_{\ell-1} \) coding the truth assignment \( X(0), \cdots, X(\ell-1) \) to \( A \).

For each subformula \( B \) of \( A \) substitute \( B \) for the corresponding atom \( p^Z_i \), where \( Z(i) \) codes the truth assignment to \( B \). In particular, substitute \( A \) for \( p^Z_0 \).

The resulting formula has the form \( \text{Eval}' \supset A \), where \( \text{Eval}' \) has a short Frege proof. Thus we obtain a \( f+ \) proof of \( A \) which is polynomial in the length of the \( g \) proof \( U_0 \) of \( A \). \( \square \)

**Strongly Uniform Reducts**

We can strengthen the definition of uniform reduct to obtain the notion of *strongly uniform reduct* of \( f \) as follows:
Definition:

\[ SU_f = \{ \varphi(\vec{X}) \in \Sigma^B_0 : \text{there is a polytime function that takes } \vec{n} \text{ to an } f\text{-proof of } \varphi(\vec{X})[\vec{n}] \} \]

where polytime means time \((\Sigma n_i)^{O(1)}\).

We can strengthen Theorem 1 for the case of strongly uniform reducts by replacing “simulates” by “p-simulates”. If \( f \) p-simulates \( g \) then there is a polytime algorithm which translates \( g \)-proofs to \( f \)-proofs, whereas if \( f \) merely simulates \( g \), then the poly-expanded \( f \)-proof exists, but there is no guarantee it can be found in polytime.

**Theorem 2:** \( SU_{f+} = \text{TRUE}_{\Sigma^B_0} \) iff \( f+ \) p-simulates every proof system.

The proof is obtained from the proof of Theorem 1 by noticing that we can efficiently construct the substitution Frege proofs involved from the \( f+ \) proofs of \( \text{Sound}_g[\vec{n}] \). □

**Remark** As far as we know, an optimal proof system might exist even though \( \text{NP} \neq \text{coNP} \).

**References**

[Bec05] Arnold Beckmann. Uniform proof complexity. *to appear in J. Logic and Computation*, 2005.

[CN] Stephen Cook and Phuong Nguyen. *Foundations of Proof Complexity: Bounded Arithmetic and Propositional Translations*. Book in progress (see authors’ web pages).

[Coo05] Stephen Cook. Theories for Complexity Classes and their Propositional Translations. In Krajíček J, editor, *Complexity of computations and proofs*, pages 175–227. Quaderni di Matematica, 2005.

[KP89] J. Krajíček and Pavel Pudlak. Propositional proof systems, the consistency of first order theories and the complexity of computations. *J. Symbolic Logic*, 54:1063–79, 1989.

[Kra95] J. Krajíček. *Bounded Arithmetic, Propositional Logic and Computational Complexity*. Cambridge University Press, 1995.