Closure Properties of Weak Systems of Bounded Arithmetic

Antonina Kolokolova

University of Toronto & Mathematical Institute, Prague
kol@cs.toronto.edu

Abstract. In this paper we study the properties of systems of bounded arithmetic capturing small complexity classes and state conditions sufficient for such systems to capture the corresponding complexity class tightly. Our class of systems of bounded arithmetic is the class of second-order systems with comprehension axiom for a syntactically restricted class of formulas $\Phi \subset \Sigma_1^B$ based on a logic in the descriptive complexity setting. This work generalizes the results of [8] and [9][1].
We show that if the system 1) extends V_0 (second-order version of $I\Delta_0$), 2) Δ_1-defines all functions with bitgraphs from Φ, and 3) proves witnessing for all theorems from Φ, then the class of Σ_1^B-definable functions of the resulting system is exactly the class expressed by Φ in the descriptive complexity setting, provably in this system.

1 Introduction

There has been a lot of research in descriptive complexity and bounded arithmetic, as well as their connections with complexity theory. However the question of direct relationship between these two fields did not receive much attention. The language of bounded arithmetic is richer than that of many logics, but often logics capture complexity classes over languages that include some arithmetic predicates (order, plus and times, or, equivalently, BIT predicate).

Bounded arithmetic studies the complexity of proving properties of these classes of formulas, whereas descriptive complexity is concerned with their expressive power. The most important distinction between different systems of bounded arithmetic is the strength of their induction (or comprehension) axiom schemes. This leads to the following question: how does the expressive power of the class of formulas in the induction axioms of a system relate to the power of the resulting system? In which cases the formulas in the comprehension are more complex than the provably total functions of a system and under which conditions their complexity coincides?

In this paper, we discuss properties under which the complexity of formulas in comprehension axioms and of provably total functions of a system of arithmetic is the same. Our approach is geared towards feasible complexity classes, those

[1] More detailed presentation of most of this work can be found in my PhD thesis, [17], available on ECCC.

L. Ong (Ed.): CSL 2005, LNCS 3634, pp. 369–383, 2005.

between $\mathtt{P}$ and $\mathtt{DLOGTIME}$ (uniform $\mathtt{AC}^0$). Restricting our attention to small classes allows us to use definability by $\mathtt{NP}$ predicates (bounded Σ_1) for the definition of capture in the bounded arithmetic setting: we consider exactly the functions with bitgraphs represented by $\mathtt{NP}$ predicates that are provably total in our systems. By Fagin's theorem [12], $\mathtt{NP}$ predicates are representable by second-order existential formulas, so the formula classes we consider here are subsets of second-order existential formulas.

Traditionally, functions are introduced by their recursion-theoretic characterization (see [4] for the original such result or [26]), but since we are trying to relate the expressive power of the formulas in comprehension and complexity of functions, we introduce function symbols by setting their bitgraphs to be formulas from the comprehension scheme.

Let C be a complexity class. Suppose that Φ_C is a class of (existential second-order) formulas that captures C in the descriptive complexity setting. We define a theory of bounded arithmetic $V\text{-}\Phi_C$ to be Robinson's Q together with comprehension over bounded Φ_C. The following is an informal statement of our main result:

Claim: *Let $\mathtt{AC}^0 \subseteq C \subseteq \mathtt{P}$. Suppose that Φ_C is closed under first-order operations provably in $V\text{-}\Phi_C$ (1). Also, suppose that for every $\phi(\bar{x}, \bar{Y}) \in \Phi_C$, if $V\text{-}\Phi_C \vdash \phi$ then there is a function F on free variables of ϕ which is computable in C and witnesses existential quantifiers of ϕ (2). Then the class of provably total functions of $V\text{-}\Phi_C$ is the class of functions computable in C.*

It may seems that the second condition, that is witnessing for the Φ_C theorems, is almost a restatement of the result itself. However, the class Φ_C can be very small, with definition of one complete problem for the class (for example transitive closure). Then the second condition states that if this small set of theorems can be witnessed, then all functions from that complexity class are provably total in the system.

For conventional systems of bounded arithmetic, such as ones considered by Clote and Takeuti in [3], it was shown that the class of provably total functions of a system coincides with the function class in the complexity-theoretic sense. Under our conditions this is provable within the system itself, so more work is needed to prove the conditions, but the result is stronger. We hope that our framework can be useful for proving independence results for weak theories of arithmetic.

Examples of systems that provably capture complexity classes are V_1-Horn capturing $\mathtt{P}$ from [7, 8], V-Krom capturing $\mathtt{NL}$ from [9] and V^0 capturing $\mathtt{AC}^0$ from [6]. As an example of a similar system that captures a complexity class, but not (known to be) provably, we present a system of arithmetic V-SymKrom corresponding to symmetric logspace ($\mathtt{SL}$), based on symmetric second-order 2-CNF formulas (with $\oplus$ instead of $\vee$ between literals). This system can prove that its class of provably total functions is the $\mathtt{AC}^0$ closure of $\mathtt{SL}$ functions. By the recent Reingold's result [22], $\mathtt{SL} = \mathtt{L}$ and so symmetric 2-SAT is solvable in logspace; therefore, $\mathtt{AC}^0(\mathtt{SL}) = \mathtt{SL} = \mathtt{L}$. However, this proof, and even the proof that $\mathtt{SL}$ is closed under complementation by Nisan and Ta-Shma [20], rely on algebraic properties on expander graphs. In their current form, these proofs are not formalizable using SL-reasoning: to talk about algebra, we need at least polynomial time. It is a very interesting open question whether there is a combinatorial version of Reingold's proof that is formalizable in a system for L, and whether our theory for SL is fully conservative over a system for L.

2 Descriptive Complexity Framework

The name "descriptive complexity" refers to the study of expressive power of logics: fixing a formula, we look at the complexity of evaluating this formula on different finite structures. It is more common to call this area "finite model theory"; however, here we stay with the term "descriptive complexity" to emphasize the complexity theory connection and the richness of the assumed vocabulary. Please see [11], [16], and [18] for the background.

Following [16], we consider logics over the vocabulary $\tau = \{\min, \max, +, \times, \leq\}$ (we do not include BIT operator since it can be defined from $+, \times$ in the weakest of our systems; see [6] for details). For many results it is sufficient to assume only the presence of order and successor relations in the vocabulary (these are the assumptions of [13, 14]); however it is more convenient to work with a vocabulary containing all basic arithmetic operations. We refer to structures where the arithmetic symbols of the vocabulary get the standard interpretation as "arithmetic structures". The way we connect logics with complexity classes is stated in this definition (following [18]):

Definition 1 (Capture by a logic). *Let C be a complexity class, L a logic and K a class of finite structures. Then L captures C on K if*

1. *For every L-sentence ϕ and every $\mathcal{A} \in K$, testing if $\mathcal{A} \models \phi$ with ϕ fixed and an encoding of $\mathcal{A}$ as an input can be done in C.*
2. *For every collection K' of structures closed under isomorphism, if this collection is decidable in C then there is a sentence $\phi_{K'}$ of L such that $\mathcal{A} \models \phi_{K'}$ iff $\mathcal{A} \in K'$, for every $\mathcal{A} \in K$.*

For our purposes, we fix K to be the arithmetic structures. In particular, the universe of a structure is always considered to be $\{0, \ldots, n-1\}$.

Many capture results are obtained by extending first-order logic with additional operators, such as fixed-point operators. We find it more convenient to work with restrictions of second-order logics rather than extensions of first-order. However, in many cases we can switch to the extended first-order logic framework by adding a defining axiom for a new operator, where the defining axiom is a second-order formula. We use this for theories of non-deterministic logspace and symmetric logspace (NL and SL), in order to introduce respective transitive closure operators.

Definition 2. *We will use the term* restricted $SO\exists$ *to refer to formulas of the form*

$$\exists P_1 \ldots P_k \forall x_1 \ldots x_l \psi(\bar{P}, \bar{x}, \bar{a}, \bar{Y}), \tag{1}$$

where k, l are constants, and ψ is a (sub)class of CNF closed under conjunction. Here, when defining a subclass of CNF we treat only the quantified second-order variables $\bar{P}$ as literals.

Note that there are no occurrences of existential first-order quantifiers in restricted $SO\exists$ formulas. This is because even when the class of ψ is restricted to 2CNF with at most one occurrence of a positive literal, with presence of an existential quantifier it is possible to capture all of $SO\exists$ [13, 14]. Universal first-order and quantifier-free formulas are restricted $SO\exists$.

Schaefer's theorem ([23]) presents several restrictions on CNF that correspond to different complexity classes. Grädel in [13, 14] described how to use some of them to capture complexity classes by restricted second-order formulas. Here we use systems based on the following restrictions of ψ:

Definition 3. *A formula $\psi(\bar{x}, \bar{P}, \bar{a}, \bar{Y})$ is* Horn *with respect to the second-order variables $P_1, ..., P_k$ if ψ is quantifier-free in conjunctive normal form and in every clause there is at most one positive literal of the form $P_i(\bar{x})$. It is* Krom *with respect to $\bar{P}$ if ψ is a CNF with at most two occurrences of a P-literal per clause. It is* SymKrom *if it is Krom with $\oplus$ instead of $\vee$ in every clause (so every clause is of the form $(\phi_i \rightarrow L_i \oplus L'_i)$, where the only P-literals are L_i and L'_i).*

Following Grädel, we can define classes $SO\exists$ Horn and $SO\exists$ Krom and $SO\exists$ SymKrom as restricted $SO\exists$, in which ψ is, respectively, Horn, Krom and SymKrom with respect to $\bar{P}$.

The following descriptive complexity characterizations provide classes of formulas on which our systems can be based. However, not all of them result in systems tightly capturing the corresponding complexity class.

Over arithmetic structures,

- First-order logic captures uniform $\mathtt{AC}^0$ ([1, 15]).
- Second-order existential logic captures `NP` ([12]), and in general levels of SO hierarchy correspond to levels of PH ([24]).
- Second-order Horn, Krom and SymKrom capture `P`, `NL` and `SL`, respectively ([13, 14]).

In case of restricted second-order formulas, the formula evaluation direction of the capture proof consists of the following steps. First, the formula is brought into propositional form by making a copy of its quantifier-free part for every possible tuple of values of quantified first-order variables. Then first-order terms and free second-order terms are evaluated. Second-order terms of the form $P_i(t(\bar{x}))$, where P_i is quantified and $t(\bar{x})$ is a term, are assigned propositional variables so that $P_i(t(\bar{x}))$ and $P_i(t'(\bar{x}))$ are assigned to the same variable whenever $t(\bar{x})$ evaluates to the same value as $t'(\bar{x})$, on possibly different tuples $\bar{x}$. Now the problem is reduced to testing satisfiability of the resulting propositional formula.

3 Bounded Arithmetic Framework

In descriptive complexity, a language in the traditional complexity theory setting is thought of as interpretations of a unary predicate X (viewed as a binary string) in a set of structures. A class of recursively enumerable languages then naturally corresponds to a class of formulas: each language in the class corresponds to a formula which has, as its set of models, the structures with X interpreted as strings from the language. In the bounded arithmetic setting, the relationship with complexity classes is slightly different. Here, we consider representations of languages in the standard model of arithmetic $\mathbb{N}_2$ (two-sorted $\mathbb{N}$). So instead of a set of structures with one predicate getting different interpretation we are talking about one fixed structure and different (second-order) elements of it satisfying the formula.

Definition 4 (Representation). *A formula $A(X)$* represents *a language L if $L = \{w(S) | \mathbb{N}_2 \models A(S)\}$, where w is some encoding of strings. More generally, $A(\bar{x}, \bar{Y})$ represents a relation $R(\bar{x}, \bar{Y})$ which holds on $\bar{x}, \bar{Y}$ iff $\mathbb{N}_2 \models A(\bar{x}, \bar{Y})$. A class of formulas Φ represents a complexity class* **C** *iff every relation R from* **C** *is representable by a formula from Φ, and every formula from Φ can be evaluated within* **C**.

This notion is parallel to the notion of "capture" from descriptive complexity (see definition 1); essentially, they have the same meaning of describing the expressive power of formulas. But the notion of "capture" we will be using for systems of bounded arithmetic will be quite different.

The language of our systems of arithmetic is $\mathcal{L}_A^2 = \{0, 1, +, \cdot, | \ |; <, =, \in\}$, a natural second-order extension of the language of Peano Arithmetic $\mathcal{L}_A = \{0, 1, +, \cdot; <, =\}$. Let $\mathbb{N}_2$ be a standard structure with natural numbers and finite sets of natural numbers in the universe; our first-order objects (denoted by lower-case letters) are natural numbers; second-order objects (denoted by upper-case letters) are binary strings or, equivalently, (finite) sets of numbers. Treating a second-order variable X as a set, its upper bound ("length") $|X|$ is defined to be the largest element $y \in X$ plus one, or 0 if X is an empty set.

Arithmetic terms are constructed using $+$ and $\times$ from first-order variables, constants 0 and 1, and terms of the form $|X|$ where X is a second-order variable. The atomic formulas of $\mathcal{L}_A^2$ have one of the forms $s = t, s \leq t, t \in X$, where s and t are terms and X is a second-order variable. We usually write $X(t)$ instead of $t \in X$. Formulas are built from atomic formulas using the propositional connectives $\wedge, \vee, \neg$, the first-order quantifiers $\forall x, \exists x$ and the second-order quantifiers $\forall X, \exists X$.

Bounded first-order quantifiers get their usual meaning: $\forall x \leq t \phi$ stands for $\forall x (x \leq t \rightarrow \phi)$ and $\exists x \leq t \phi$ stands for $\exists x (x \leq t \wedge \phi)$. Second-order quantified variables are strings of bounded length; the notation $\exists Z \leq b$ corresponds to $\exists Z \ |Z| \leq b$.

Definition 5. *Σ_0^B and Π_0^B both denote the class of bounded formulas with no second-order quantifiers. We define inductively Σ_{i+1}^B as the least class of formulas containing Π_i^B and closed under disjunction, conjunction, and bounded existential second-order quantification. The class Π_{i+1}^B is defined dually. We use notation $\Sigma_0^B(\Phi)$ to refer to the closure of Φ under first-order operations: that is, under $\vee, \wedge, \neg$ and bounded first-order $\forall$ and $\exists$.*

3.1 Translation

Let Φ be a descriptive logic over a vocabulary τ. For every $\phi \in \Phi$, we can define a translation ϕ^* into $\mathcal{L}_A^2$ with the following properties:

1. Every interpreted symbol from τ that occurs in $\mathcal{L}_A^2$ gets the standard interpretation, e.g., successor becomes $+1$, min becomes 0, etc.
2. Translate max as n for a free variable n. For every quantified first-order variable, set $n+1$ (more generally, a polynomial of n) as a bound. Note that then $|X| = n+1$ for a unary second-order predicate.
3. Translate uninterpreted relational symbols of τ occurring in ϕ as free second-order variables of ϕ^*. If a variable is k-ary, use a pairing function to encode the relational symbol as a unary second-order variable. Then any occurrence of $R(x_1, \ldots, x_k)$ becomes $R^*(\langle x_1, \ldots, x_k \rangle)$, where $\langle x_1, \ldots, x_k \rangle$ is a value obtained by applying the pairing function to $x_1, \ldots, x_k$.

Under this translation, a restricted second-order formula becomes a restricted Σ_1^B formula with the same restriction on the quantifier-free part. The resulting Φ^* represents in the standard model the same complexity class as is captured by Φ in the descriptive complexity setting.

Table 1. The 2-BASIC axioms

B1: $x+1 \neq 0$	B2: $x+1 = y+1 \to x = y$	B4: $x + (y+1) = (x+y)+1$
B3: $x+0 = x$	B5: $x \cdot 0 = 0$	B6: $x \cdot (y+1) = (x \cdot y) + x$
B7: $0 \leq x$	B9: $x \leq y \wedge y \leq z \to x \leq z$	B10: $(x \leq y \wedge y \leq x) \to x = y$
B8: $x \leq x + y$	B11: $x \leq y \vee y \leq x$	B12: $x \leq y \leftrightarrow x < y+1$
L1: $X(y) \to y < \lvert X \rvert$	L2: $y+1 = \lvert X \rvert \to X(y)$	B13: $x \neq 0 \to \exists y(y+1 = x)$

3.2 Systems of Bounded Arithmetic

Now, for a set of formulas Φ, a system V-Φ is axiomatized by 2-BASIC axioms listed in table above together with a comprehension scheme of the form

$$\exists Z \leq b \forall i < b (Z(i) \leftrightarrow \phi(i, \bar{a}, \bar{X})), \qquad (\Phi\text{-comp})$$

where $\phi \in \Phi$.

To agree with the common notation, we abbreviate V-Σ_i^B as V^i, $i \geq 0$. These theories are axiomatized by the 2-BASIC together with a comprehension scheme for Σ_i^B formulas. For $i \geq 1$, V^i is equivalent to the first-order theory S_2^i by RSUV isomorphism [21, 25]. The system V^0 corresponds to the complexity class uniform $\mathtt{AC}^0$.

4 Definability in V-Φ

4.1 Basic Properties of V^0 and V-Φ

The system V^0 is robust enough to prove many natural properties. In particular, induction on the length of string (and thus on Σ_0^B combinations of Φ) is a theorem of V-Φ extending V^0. Also, V^0 proves properties of the pairing function and simultaneous comprehension over several variables, resulting in an array (so several existential second-order quantifiers can be treated as one). We use $P^{[b]}$ to denote the "b-th row" when P is being used as a 2-dimensional array. If $\phi(P)$ is a formula with no occurrence of $|P|$, then $\phi(P^{[b]})$ is obtained from $\phi(P)$ by replacing every atomic formula $P(t)$ by $P(b, t)$.

The following property, Replacement, plays a major role in our definability proofs. It is a theorem for V^1 and stronger theories, however weaker theories do not prove full Σ_1^B replacement under cryptographic assumptions by [10]. For our purpose it is sufficient to prove it for restricted Σ_1^B formulas.

Lemma 1 (Replacement). *Let Φ be a class of restricted Σ_1^B formulas. Then for every formula $\exists \bar{P}\phi(y, \bar{P}) \in \Phi$, where ϕ can have additional free variables, V-Φ proves*

$$\forall y < t \exists \bar{P}\phi(y, \bar{P}) \leftrightarrow \exists \bar{P} \forall y < t \phi(y, \bar{P}^{[y]}) \qquad \text{(Replacement)}$$

where $\bar{P}^{[y]}$ is $P_1^{[y]}, ..., P_k^{[y]}$.

Proof. The proof is a generalization of a proof of Replacement in [8]. Here we are using the lack of existential first-order quantifiers and closure under conjunctions of the quantifier-free parts of Φ-formulas.

4.2 Function Classes

Complexity classes are defined as classes of relations. This is also the interpretation for the descriptive complexity setting. But in bounded arithmetic the measure of the power of a theory is the complexity of the corresponding functions. So we use relations as graphs to define number functions and as bit graphs to define string functions. The following definition is very general, but sometimes does not produce a robust function class: for example, there is nothing in this definition that would force the functions to be closed under composition. In order to make the function classes defined this way meaningful, we will need additional restrictions.

Definition 6. *Let C be a complexity class. We define the corresponding class FC of functions of C as follows: A string function $F : \mathbb{N}^k \times (\{0,1\}^*)^l \to \{0,1\}^*$ is in FC iff there is a relation R in C and a polynomial p such that $F(\bar{x}, \bar{Y})(i) \leftrightarrow i < p(\bar{x}, |\bar{Y}|) \wedge R(i, \bar{x}, \bar{Y})$ for all $i \in \mathbb{N}$. A number function $f(\bar{x}, \bar{Y})$ is in the class FC if there is a string function in $F(\bar{x}, \bar{Y}) \in FC$ such that $f(\bar{x}, \bar{Y}) = |F(\bar{x}, \bar{Y})|$. If formula class Φ represents C, then R can be replaced by a formula $\phi \in \Phi$ representing R.*

For string functions, we are only concerned with the bits with indices smaller than $p(\bar{x}, \bar{Y})$. Therefore, a string corresponding to the value of a function will be of length less than $p(\bar{x}, \bar{Y})$. In particular, by the length axioms, all bits with indices larger than $p(\bar{x}, \bar{Y})$ are 0.

This definition of FC does not directly impose any "robustness" conditions such as closure under function composition. To allow for that, we define an $\mathtt{AC}^0$ closure of FC as follows.

Definition 7. *A (string) function $F(\bar{x}, \bar{Y})$ is $\mathtt{AC}^0$ reducible to a set of function symbols $\mathcal{L}$ (denoted $F \in \mathtt{AC}^0(\mathcal{L})$) iff there is a sequence $F_1 \dots F_n$ of string functions such that $F_n = F$ and F_i is in $\Sigma_0^B(\mathcal{L} \cup \{F_1 \dots F_{i-1}\})$ for $i = 1, \dots, n$. If for any $F \in \mathtt{AC}^0(\mathcal{L})$, $F \in \mathcal{L}$ we say that $\mathcal{L}$ is closed under $\mathtt{AC}^0$ reductions.*

In case FC is definable by formulas from Φ, the definition naturally generalizes to $\mathtt{AC}^0(\Phi)$.

Definition 8. *A relation $R(\bar{x}, \bar{Y})$ is Δ_1^B-definable in V-Φ iff there exist formulas $\phi, \tilde{\phi} \in \Sigma_1^B$ such that $R(\bar{x}, \bar{Y})$ is represented by $\phi(\bar{x}, \bar{Y})$ and $V\text{-}\Phi \vdash \phi(\bar{x}, \bar{Y}) \leftrightarrow \neg\tilde{\phi}(\bar{x}, \bar{Y})$. A string function F is Σ_1^B-definable in V-Φ if it has a defining axiom $Z = F(\bar{x}, \bar{Y}) \leftrightarrow \phi(Z, \bar{x}, \bar{Y}))$, with $\phi \in \Sigma_1^B$ such that $V\text{-}\Phi \vdash \forall \bar{x} \forall \bar{Y} \exists ! Z \phi(Z, \bar{x}, \bar{Y}))$.*

By the second-order version of Parikh's theorem (see [6]), we can use Σ_1^B-definability and Σ_1-definability interchangeably. Also, Δ_1^B-definable relations and Σ_1^B-definable boolean functions are the same (consider characteristic functions of predicates).

Using definition 8, we can state the definition of "capture" in the bounded arithmetic setting. This gives us a way of measuring the power of a system of arithmetic.

Definition 9 (Capture in bounded arithmetic). *A system of arithmetic T* captures *a complexity class C if the class of Σ_1^B-definable functions of T is exactly FC. That is, FC is the class of functions representable by Σ_1^B formulas that are provably total in T.*

Note that this is quite different from the descriptive complexity notion of "capture" from definition 1: descriptive complexity "captures" is bounded arithmetic "representable". The reason we are using the same word is that in both cases we are relating a logic (system of arithmetic) and a complexity class; "capture" here is a generic name for such a connection.

4.3 Properties

The first property that we consider is (provable) closure under $\mathtt{AC}^0$ reductions. We emphasize the provability part here: in the previous work, e.g., by Clote and Takeuti [2], the fact that the classes in question were closed under complementation was used but not proven within the system.

*Property 1 (***Closure***).* Let Φ represent a complexity class C and let FC be as in definition 6. Then the *closure property* holds if Φ is closed under $\mathtt{AC}^0$ reductions. In particular, FC is closed under composition and substitution of a term for a variable. In addition, Φ is *strongly* closed if for every $\phi^* \in \Sigma_0^B(\Phi)$ there exists $\phi \in \Phi$ such that $V\text{-}\Phi \vdash \phi^* \leftrightarrow \phi$.

If this property holds, the corresponding C must be closed under complementation and Φ extends Σ_0^B (that is, defines all of first-order). For some Φ, notably restricted Σ_1^B, it is not syntactically true that $\Sigma_0^B \subseteq \Phi$, but it can be proved that for any Σ_0^B formula there is an equivalent formula of Φ.

In order for a logic to translate into a "nice" system of arithmetic, the logic has to be in some sense "natural". That is, its properties such as closure under composition and complementation have to be provable using only simple concepts. Moreover, it should be easy to verify whether a given formula holds on a structure. More formally, we need the following property:

*Property 2 (***Constructiveness).** Let Φ be a class of restricted Σ_1^B formulas, and let Φ represent C. This Φ has the *constructiveness property* if the following two conditions hold. Firstly, every $\phi \in \Phi$ defines a relation R that is Δ_1^B-definable in V-Φ, with ϕ being its Σ_1^B definition. Secondly, there are witnessing functions $\bar{F}$ with bit graphs in $\Sigma_0^B(\Phi)$ such that $\bar{F}(\bar{a}, \bar{Y})$ witness the existential quantifiers of the prenex form of $\phi \vee \tilde{\phi}$.

If, additionally, Φ is strongly closed, that is, has property 1, then the conclusion of the constructiveness property can be stated simpler as follows.

Property 2' (**Strong constructiveness**) For every $\phi \equiv \exists \bar{P} \psi(\bar{P}, \bar{a}, \bar{Y}) \in \Phi$ such that V-$\Phi \vdash \phi$ there are functions $\bar{F}$ witnessing $\bar{P}$ such that bitgraphs of $\bar{F}$ are in Φ.

It is enough to consider ϕ-theorems of V-Φ because if Φ is closed, then $\tilde{\phi} \in \Phi$ and so is $\phi \vee \tilde{\phi}$. Also, the assumption that bitgraphs of $\bar{F}$ are in $\Sigma_0^B(\Phi)$ becomes bitgraphs $\in \Phi$.

Sometimes we use the term "weak constructiveness" to refer to the original constructiveness property, and "strong constructiveness" for the second version.

4.4 Main Results

Now we are ready to state the main theorem of this paper.

Theorem 1 (Definability theorem). *Suppose that Φ is restricted Σ_1^B or Σ_0^B, constructive, and represents a complexity class C. Then all functions from FC are Σ_1^B-definable in V-Φ and all Σ_1^B-definable functions of V-Φ are in* $\mathtt{AC}^0(FC)$.

Suppose, additionally, that Φ is strongly closed. In this case, the class of Σ_1^B-definable functions of V-Φ coincides with FC provably in V-Φ.

We will refer to the first statement as "weak definability" and the second statement as "strong definability".

The proof of this theorem consists of two parts. The part that FC is Σ_1^B-definable in V-Φ follows by the fact that we have comprehension for $\Sigma_0^B(\Phi)$-formulas, which gives us replacement for both ϕ and its Σ_1^B negation.

The second part, which we call the *generalized witnessing theorem*, is used to show that the class of witnessing functions for ϕ-formulas is $\mathtt{AC}^0(FC)$.

Theorem 2 (Generalized witnessing theorem). *Let Φ be a class of restricted Σ_1^B formulas representing C. Suppose that Φ is constructive. Then Σ_1^B-theorems of V-Φ can be witnessed by functions from $\mathtt{AC}^0(FC)$ provably in V-Φ. That is, if V-$\Phi \vdash \exists Z\phi(\bar{x},\bar{Y},Z)$, where $\phi \in \Sigma_1^B$, then there is a string function $F(\bar{x},\bar{Y})$ in $\mathtt{AC}^0(FC)$ such that*

$$V\text{-}\Phi, AX(F) \vdash \phi(\bar{x},\bar{Y},F(\bar{x},\bar{Y})),$$

where $AX(F)$ is a defining axiom for F. If Φ is strongly closed and constructive, then V-Φ proves that the defining axiom for F is equivalent to a formula from Φ.

The witnessing theorem looks similar to the constructiveness property, but they talk about different classes of formulas. Whereas constructiveness is concerned with witnessing an existential quantifier in a $\phi \in \Phi$ (or finding a counterexample to ϕ), the witnessing theorem describes the power of a system in terms of the strength of Σ_1^B-theorems that the system in question can prove.

The theorem 2 is a generalization of the witnessing theorem for V^0 as presented in [6] (hence the name "Generalized witnessing"). The proof uses proof-theoretic techniques. Taking a Σ_1^B theorem of V-Φ, we analyze its anchored proof in a second-order version of quantified Gentzen calculus LK^2 and prove, by induction on the structure of the proof, that in every line existential quantifiers can be witnessed by the functions of given complexity. To ensure that every line in the proof has only Σ_1^B formulas, we replace the comprehension axiom of V-Φ by a statement of the form $\exists Z < t\forall i \leq t(\phi(i) \wedge Z(i)) \vee (\tilde{\phi}(i) \wedge \neg Z(i))$, $\phi \in \Phi$, where $\tilde{\phi}$ is a Σ_1^B formula equivalent to the negation of ϕ, provided by the constructiveness property. This gives us the base case (witnessing for the axioms). The witnesses in the rest of the cases are $\mathtt{AC}^0$ combinations of witnesses in the previous steps.

Note that if the conditions do not hold, then the class of witnessing functions can be smaller than representable by formulas in the comprehension axiom. An example of that is the theory V^1, with comprehension over $\mathtt{NP}$ predicates. By the second-order version of Buss' witnessing theorem [6, 26], the class of Σ_1^B-definable functions of V^1 is $\mathtt{P}$. But not every Σ_1^B formula is Δ_1^B-definable in V^1. Moreover, even if $\mathtt{NP} = co\mathtt{NP}$ and for every Σ_1^B formula there is an equivalent Π_1^B formula, it might not be the case that these equivalences are provable in V^1.

5 Applications of the Definability Theorem

In this section we restate several previously known capture results in our framework. Three such examples when the strong case of Theorem 1 applies are V^0 itself, V_1-Horn and V-Krom. Below, we show that these theories are built on classes of formulas satisfying our two properties.

Example 1([5, 6, 26]) Functions bit-definable by Σ_0^B formulas in V^0 are $\mathtt{AC}^0$ functions, and Σ_0^B formulas correspond to the first-order logic which captures $\mathtt{AC}^0$ in the descriptive sense ([1]). The constructiveness property is satisfied trivially, since Σ_0^B is closed under complementation syntactically and there are no quantifiers to witness. It was shown in [5, 26] that $\mathtt{AC}^0$ functions are closed under composition and thus under $\mathtt{AC}^0$ reductions. Therefore, theorem 1 applies, so the class of Σ_1^B-definable functions of V^0 is $F\mathtt{AC}^0$.

Example 2 ([7, 8]) The class of Σ_1^B-Horn formulas comes from $SO\exists$-Horn formulas capturing $\mathtt{P}$ in the descriptive setting. The resulting system V_1-Horn defines polynomial-time functions by Σ_1^B-Horn formulas, and is equivalent in power to Zambella's $\mathtt{P}$-def (and thus PV). In this case, the properties hold with $\Phi = \Sigma_1^B$-Horn and $FC = FP$. So by the definability theorem Σ_1^B-definable functions of V_1-Horn are precisely polynomial-time functions. The bulk of work is a formalization of the satisfiability algorithm for propositional Horn formulas, which is needed already to prove closure of Σ_1^B-Horn formulas under complementation. This algorithm is constructive: a satisfying assignment (or, equivalently, values for quantified second-order variables) is obtained as part of the algorithm (the value $T^{[a]}$ in the description of RUN). This gives the constructiveness property.

Example 3([9]) Now take the class of Σ_1^B-Krom formulas, a translated version of Grädel's $SO\exists$-Krom (second-order 2CNF). It is possible to formalize Immerman-Szelepcsényi's proof of closure of $\mathtt{NL}$ under complementation in the resulting theory V-Krom ([9]). Also, proving that transitive closure function is Σ_1^B-definable in V-Krom results in a proof of constructiveness for V-Krom: values for quantified second-order variables are expressed as Σ_0^B combination of transitive closure function calls.

The next example, a system of arithmetic for $\mathtt{SL}$, presents a case when we were not able to prove the strong version of the properties; this led to the formulation of the weaker properties.

6 Weak Case of the Definability Theorem

A class of Σ_1^B-SymKrom formulas is very similar to Σ_1^B-Krom, except it is based on symmetric 2CNF (that is, 2CNF with XOR instead of disjunctions). From the same Grädel's paper as before, [14], we know that $SO\exists$-SymKrom captures $\mathtt{SL}$. We define V-SymKrom to be V-Φ with $\Phi \equiv \Sigma_1^B$-SymKrom.

It seems that showing that a system V-SymKrom would capture $F\mathtt{SL}$ should be straightforward. However, the methods used to prove closure of $\mathtt{SL}$ under complementation (Nisan and Ta-Shma, [20]), and, recently, that $\mathtt{SL} = \mathtt{L}$ (Reingold, [22]) use properties of expander graphs and rely on algebraic methods for the proofs. But those are not known to be formalizable in less complexity than $\mathtt{P}$. By Reingold's result, the class of Σ_1^B-definable functions of V-SymKrom is thus all logspace functions, but this is not known to be provable in V-SymKrom itself, as opposed to the cases of $\mathtt{AC}^0, \mathtt{NL}$ and $\mathtt{P}$. It might still be possible that such a theory for $\mathtt{SL}$ is not fully conservative over a theory for $\mathtt{L}$.

6.1 Symmetric Transitive Closure

To simplify proofs, we introduce symmetric transitive closure operator by the following axiom:

$$STC_{x,y}\phi(x,y,\bar{a},\bar{Y})[a,b,n] \leftrightarrow \forall R(CondS(\phi,R,n) \to R(a,b)), \qquad \text{(AxSTC)}$$

where

$$CondS(\phi,R,n) \equiv \forall x,y,z < n(R(x,x) \wedge (\phi(x,y) \to (R(y,z) \leftrightarrow R(x,z))))$$

Note that if ϕ is quantifier-free except for bounded existential first-order quantifiers, then the negation of the $STC_{x,y}\phi(x,y)[a,b,n]$ defining axiom is equivalent to a Σ_1^B-SymKrom formula. Therefore, V-SymKrom proves induction on Σ_0^B combinations of STC functions.

By the same reasoning as for V-Krom in [9], STC defined in this manner is reflexive, transitive and robust against adding an edge on the left versus on the right (that is, conditions with $\phi(x,y) \to (R(x,z) \leftrightarrow R(y,z))$ and $\phi(y,z) \to (R(x,z) \leftrightarrow R(x,y))$ are equivalent). It is also provable in V-SymKrom that STC is symmetric: $STC(a,b,n) \leftrightarrow STC(b,a,n)$.

To see that $V^0 \subset V$-SymKrom, we encode a first-order formula as a graph and apply the STC operator to it. A first-order existential quantifier in $\exists z < n\psi(z)$ is simulated by STC applied to the graph with an edge relation defined by $E(x,y) \leftrightarrow \neg\psi(x) \wedge y = x+1$. That is, a graph is a path from vertex 0 to vertex n with every edge $(z, z+1)$ labeled $\neg\psi(z)$; if $\psi(z)$ holds for some z_0 then the edge (z_0, z_0+1) is absent so the start of the path and the end of it are disconnected. Similarly, a first-order universal quantifier is encoded by a graph with $E(x,y)$ such that $E(s,u) \leftrightarrow E(u,t) \leftrightarrow \neg\psi(u)$. This construction is applied for every block $\exists z < n\forall u < n\psi(z,u)$: such block is encoded as a path with every edge replaced by a "nested diamonds" gadget encoding a universal quantifier. A vertex $\langle n,n\rangle$ is reachable from the vertex $\langle 0,0\rangle$ iff $\exists z < n\forall u < n\psi(z,u)$ holds.

Now we need to show the weak constructiveness property. First, we show how to witness formulas from Σ_1^B-SymKrom using $\Sigma_0^B(STC)$. Second, we give a Σ_1^B predicate equivalent to the negation of STC and show how to witness it: since the value of every formula can be expressed using STC, this is sufficient for Δ_1^B-definability of Σ_1^B-SymKrom.

6.2 Constructing a Witness for a Σ_1^B-SymKrom Formula

Given a Σ_1^B-SymKrom formula $\phi^* \equiv \exists P\forall\bar{x} < \bar{n}\psi(P,\bar{x})$, we create a formula $\phi'(u,s,v,s')$ encoding the structure of ψ; this encoding is similar to the encoding used in [9] for Σ_1^B-Krom formulas. For every clause, $\phi'(u,s,v,s')$ says that P-literals contain terms evaluating to u and v, with s and s' being 0 if the literal is negated and 1 otherwise. A propositional version of the formula is satisfiable if the corresponding graph is bipartite, that is, $\exists R\forall u,v < b\forall s,s' < 2(\phi'(u,s,v,s') \to \neg R(u,s) \leftrightarrow R(v,s'))$. Now, to use STC to test bipartiteness we use the standard technique of "doubling" the graph, with every vertex having "even" and "odd" version and every edge connecting the literals on opposite sides. There is an odd cycle in the original graph (and thus the formula evaluates to false) iff there is a path from a vertex on one side to the same numbered vertex on the other; this can be expressed using STC. From the witness to the negation of STC we construct a value for P (all literals on the same side as the constant $\top$ are set to true).

6.3 Δ_1^B-Definability of STC

Saying that a pair (a, b) is in the symmetric transitive closure of a graph is equivalent to the statement that b is reachable from a in an undirected graph. The following Σ_0^B predicate $\textsc{ReachCond}(R, E, n+1, a)$ states that $R(x, i)$ is true iff x is at most distance i from a:

$$\forall x \leq n \forall i \leq n (R(x,0) \leftrightarrow x = a) \wedge \\ (R(x, i+1) \leftrightarrow (\exists y \leq n R(y,i) \wedge (E(y,x) \vee y = x)))$$

Let ϕ be a formula defining an edge relation of a graph. Let

$$UDist_\phi(x, y, d) \equiv STC_{(u,c),(v,c')}\alpha[(x,0),(y,d),(n,n)],$$

where $\alpha(u, c, v, c') \equiv (c' = c + 1 \wedge (\phi(u, v) \vee u = v))$. For simplicity, we assume that ϕ is represented by the corresponding graph E, and write $UDist(x, y, d)$ in that case. Then, $R(x, i) \equiv UDist(a, x, i)$ satisfies $\exists R \textsc{ReachCond}(R, E, n+1, a)$, and V-SymKrom $\vdash STC(a, b, n) \leftrightarrow \exists R \textsc{ReachCond}(R, E, n+1, a) \wedge R(b, n)$.

Now, we showed that the weak constructiveness property holds. Therefore, every $\mathtt{SL}$ function is Σ_1^B-definable in V-SymKrom and every Σ_1^B-definable function of V-SymKrom is in $\mathtt{AC}^0(F\mathtt{SL})$ provably in V-SymKrom. We know that $\mathtt{AC}^0(F\mathtt{SL}) = F\mathtt{L}$, that is every $\mathtt{AC}^0(\mathtt{SL})$ function is already computable in logspace, but this is not known to be provable in V-SymKrom. Also, just like V-Krom, V-SymKrom is finitely axiomatizable by finite set of axioms of V^0 together with comprehension over $\neg AxSTC$.

7 Conclusion

In this work we present a general framework for constructing systems of arithmetic with predefined power based on descriptive complexity results. The setback is that whereas for capture results in the descriptive complexity setting it is sufficient to have "some" proof of capture, in our bounded arithmetic framework we need an "easy" proof of capture, getting in return a "provable" capture result. It is interesting to see in which cases the complexity classes behave nicely, like $\mathtt{P}$ or $\mathtt{NL}$, and in which cases, like $\mathtt{SL}$, the proofs use concepts not (known to be) formalizable within the class itself.

A general witnessing theorem applying to slightly different types of theories was presented recently by Cook and Nguyen [19]. Their framework applies to theories equivalent to universal theories. They have a large number of applications, including different theories for $\mathtt{NL}$, $\mathtt{SL}$ and $\mathtt{P}$. However, they do not talk about provable capture.

Yet another property, uniqueness, that can be used instead of constructiveness was suggested to me by Sam Buss. This property states that for every formula from Φ there is an equivalent Σ_1^B formula with at most one witness to the quantifiers. The uniqueness property immediately implies constructiveness.

In general, it is interesting to explore the "robustness" of complexity classes such as provability of their properties. We hope that our framework provides a natural setting for such study.

Acknowledgments

I am very grateful to Stephen Cook for his supervision of my PhD thesis, which contained most of the ideas that appear in this paper. In particular, the main ideas used for the SL theory were suggested to me by him.

References

1. D. M. Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41(3):274 – 306, 1990.
2. P. Clote and G. Takeuti. Bounded arithmetic for NC, ALOGTIME, L and NL. *Annals of Pure and Applied Logic*, 56:73 – 117, 1992.
3. P. Clote and G. Takeuti. First order bounded arithmetic and small boolean circuit complexity classes. In *Feasible Mathematics*, volume II. Birkhäuser Inc., 1995.
4. A. Cobham. The intrinsic computational difficulty of functions. In Y. Bar-Hillel, editor, *Logic, Methodology and Philosophy of Science*, pages 24–30, Amsterdam, 1965. North-Holland.
5. S. Cook. Theories for complexity classes and their propositional translations. *submitted*, pages 1–36, 2004.
6. S. A. Cook. CSC 2429S: Proof Complexity and Bounded Arithmetic. Course notes, URL: "http://www.cs.toronto.edu/~sacook/csc2429h", Spring 1998-2002.
7. S.A. Cook and A. Kolokolova. A second-order system for polynomial-time reasoning based on Grädel's theorem. In *Proceedings of the Sixteens annual IEEE symposium on Logic in Computer Science*, pages 177–186, 2001.
8. S.A. Cook and A. Kolokolova. A second-order system for polytime reasoning based on Grädel's theorem. *Annals of Pure and Applied Logic*, 124:193–231, 2003.
9. S.A. Cook and A. Kolokolova. Bounded arithmetic of NL. In *Proceedings of the Nineteens annual IEEE symposium on Logic in Computer Science*, pages 398–407, 2004.
10. Stephen Cook and Neil Thapen. The strength of replacement in weak arithmetic. In *Proceedings of the Nineteens annual IEEE symposium on Logic in Computer Science*, pages 256–264, 2004.
11. H.-D. Ebbinghaus and J. Flum. *Finite model theory*. Springer Verlag, 1995.
12. R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of computation, SIAM-AMC proceedings*, 7:43–73, 1974.
13. E. Grädel. The Expressive Power of Second Order Horn Logic. In *Proceedings of 8th Symposium on Theoretical Aspects of Computer Science STACS '91, Hamburg 1991*, volume 480 of *LNCS*, pages 466–477. Springer-Verlag, 1991.
14. E. Grädel. Capturing Complexity Classes by Fragments of Second Order Logic. *Theoretical Computer Science*, 101:35–57, 1992.
15. N. Immerman. Relational queries computable in polytime. In *14th ACM Symp.on Theory of Computing, Springer Verlag (Heidelberg, FRG and NewYork NY, USA)-Verlag*, pages 147 –152, 1982.
16. N. Immerman. *Descriptive complexity*. Springer Verlag, New York, 1999.
17. A. Kolokolova. *Systems of bounded arithmetic from descriptive complexity*. PhD thesis, University of Toronto, October 2004.
18. L. Libkin. *Elements of Finite Model Theory*. Springer Verlag, 2004.
19. Phuong Nguyen and Stephen Cook. Theories for TC^0 and other small complexity classes. *submitted*, 2004.

20. Noam Nisan and Amnon Ta-Shma. Symmetric logspace is closed under complement. In *Proc. 27th Ann. ACM Symp. on Theory of Computing (STOC'95)*, pages 140–146, 1995.
21. A. Razborov. An equivalence between second-order bounded domain bounded arithmetic and first-order bounded arithmetic. In P. Clote and J. Krajiček, editors, *Arithmetic, proof theory and computational complexity*, pages 247–277. Clarendon Press, Oxford, 1993.
22. O. Reingold. Undirected ST-Connectivity in Log-Space. *Electronic Colloquium on Computational Complexity*, ECCC Report TR04-094, 2004.
23. T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, pages 216–226, 1978.
24. L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.
25. G. Takeuti. RSUV isomorphism. In P. Clote and J. Krajiček, editors, *Arithmetic, proof theory and computational complexity*, pages 364–386. Clarendon Press, Oxford, 1993.
26. D. Zambella. Notes on polynomially bounded arithmetic. *The Journal of Symbolic Logic*, 61(3):942–966, 1996.