Theories and Proof Systems for the PSPACE and the EXP-Time Hierarchy
Thesis Proposal Document

$Id: wtheories.tex,v 1.50 2005/06/10 20:26:50 alan Exp $ LaTeX'd on June 10, 2005

Alan Skelley

June 10, 2005

1 Introduction

This document is originally a working paper recording our results in progress. It is hoped that with some re-organization, addition of basic definitions, introduction and conclusion, and of course completion of the (anticipated) results, it will become a thesis. Immediately below is a short idealized introduction describing our over-all aims, and following that is an outline of what has been done and what remains.

1.1 (Idealized) Introduction

In this article we present bounded arithmetic theories and propositional proof systems corresponding to large complexity classes, including polynomial space (PSPACE), exponential time (EXP) and the levels of the nondeterministic exponential-time hierarchy. Zambella, Cook and more recent authors have used second-order (i.e., two-sorted) theories to great effect to capture the string-based computation of Turing machines more simply than with previous one-sorted theories of bounded arithmetic. Furthermore, this second-order "viewpoint" has allowed the development of theories for some very small complexity classes previously not captured with one-sorted theories due to their inherent coarseness. We extend the second-order viewpoint to higher complexity classes in a natural way by adding a third sort to represent exponentially large objects such as computations from these strong classes, outputs of unbounded exponential-time functions, or even oracles.

Another contribution of this article is to define a calculus of functions that operate on these three sorts of objects. From a complexity standpoint for the classes we are interested in, the objects are the usual binary strings, always of polynomial length; numbers, which are to be thought of as short inputs and presented in unary for the purposes of resource bounds; and finally, superstrings: exponential-length strings indexed by standard binary strings, and not counted in any resource bounds. This function calculus is very nicely suited for expressing the computational objects reasoned about by our third-order theories of bounded arithmetic; additionally, it is useful for discussing with one unified notation both polynomially-bounded functions and more exotic functions dealing with exponential-sized inputs and outputs. We define complexity classes of functions and predicates, and in each case the string functions or string predicates in these classes constitute exactly the corresponding complexity class of polynomially-bounded string functions or languages.

Finally, we give translations of certain theorems of some of these theories into quantified versions of BPLK.

1.2 Overview and Proposal for the Thesis

The document that follows is in a somewhat intermediate state. Its contents are in approximately the same order as in a future thesis, but some things are missing, and there are notes about unresolved questions or issues directly in the text. Here we take the reader through what has been done and what remains to be done. We present this as a kind of idealized table of contents. Even portions described as essentially complete should of course be understood to require polishing and in most cases the addition of more and clearer narrative.

1. **Introduction and Related Work.** To be done; can borrow heavily from research proposal and depth oral papers.

2. **The Third-Order Viewpoint.** This is basically section 2 below and addresses the language of third-order arithmetic, as well as third-order computation. This is new; although some authors have discussed higher-order computation before, it hasn't been fully addressed, particularly not with respect to defining functions in full generality as we aim to do. Previous higher-order theories for our classes were number-based, and thus did not follow the newer Zambella-Cook framework of string-based theories.

3. **Preliminaries.** To be done; this involves importing definitions of things such as BPLK from my master's thesis and papers.

4. **Third-Order Theories.** Basically section 3 below; defines the axiom schemes and main theories to be considered. These are all new. There are W_1^i and TW_1^i, analogues of U_2^i and V_2^i (see below for the intended complexity classes), as well as HW_1^0, a weaker PSPACE theory based on a recursion scheme, and TTW_1^0, an exponential-time theory which is to W_1^0 and W_1^1 as TV^0 is to V^0 and V^1.

5. **Third-Order Parikh's Theorems.** Section 4 below; this is new, although of course based on the lower-order case. Nevertheless, required some care to do properly.

6. **Basic Facts About the Theories.** This section will include basic points such as when induction is derived in a theory from comprehension, how the theories relate to lower-order theories, and so on. Section 5 below contains some of these facts in extremely condensed form; more will be added as the need arises.

7. **Replacement Schemes.** Proof of replacement in some theories. Section 6 below. As written, applies only to W_1^1 and $\Sigma_1^{\mathcal{B}}$-replacement, and must be expanded to W_1^i and $\Sigma_i^{\mathcal{B}}$-replacement. These proofs are adapted from Buss' thesis and are not really a contribution.

8. **Definability in the Theories.** Section 7 below contains our definition of third-order definability, which covers the most general case of arguments and function value of any sort. These were not obvious, especially in the case of superstring-valued functions, which are problematic as our third-order variables are unbounded. The existing definability results in that section are for ordinary string functions only, and only for W_1^1. We aim to prove that $(\text{FPSPACE}^{(\Sigma_{i-1}^{exp})^\diamond})^+$ and $(\text{FEXP}^{(\Sigma_{i-1}^{exp})^\diamond})^+$ are $\Sigma_i^{\mathcal{B}}$-definable in respectively W_1^i and TW_1^i; restricted to strings, these classes are exactly the usual complexity classes $\text{FPSPACE}^{\Sigma_{i-1}^p}$ and $\text{FEXP}^{\Sigma_{i-1}^p}$.

Although these results have lower-order analogues, they are nevertheless somewhat new; in particular, our theories are like Buss' original U_2^1 and V_2^1 in that they are "unbounded domain"; later results about U_2^i and V_2^i and the exponential-time hierarchy pertain to Razborov's "bounded domain" versions, which sidestep the problems of unbounded higher-order objects.

9. **A Universal Conservative Extension of HW_1^0.** This is basically done in section 8 below; it is somewhat nontrivial also, and new. If there is time, we plan to adapt the Cook-Thapen argument to use this open theory to show that HW_1^0 does not prove the replacement scheme, subject to a complexity assumption.

   Since we also have a recursion-theoretic characterization of FEXP$^+$, it should be possible to obtain an open theory for exponential time with only a minor change, and if time allows we will do this as well.

10. **Witnessing Theorems.** Section 10 below contains witnessing theorems for W_1^1 and HW_1^0 for the case of string functions; these are new but must be generalized to the full third-order functions, and further expanded to W_1^i and TW_1^i (to match the results about definable functions in these theories).

11. **Propositional Translations.** Section 11 below contains a translation of $\Sigma_0^{\mathcal{B}}$ theorems of W_1^1 into BPLK, which is new. We aim to generalize this translation to $\Sigma_i^{\mathcal{B}}$ theorems of W_1^i and a quantified version of BPLK.

12. **Conclusions.** To be done.

To summarize, our contributions thus far are: the third-order viewpoint and function calculus; the third-order theories; third-order Parikh's theorems; general definitions of third-order definability; special cases (string functions only; first level of hierarchy) of the definability and witnessing theorems for the theories; the universal conservative extension of HW_1^0; propositional translation into BPLK.

What remains to be done is: generalize provability of replacement schemes; generalize definability and witnessing theorems; clarify, expand and polish throughout. All the preceeding should be straightforward. Showing the conditional separation of replacement from HW_1^0 is not expected to be difficult if undertaken, nor is the universal theory for exponential time. The thorniest remaining task is to generalize the propositional translations.

2 The Third-Order Viewpoint

In this section we define and explain the features and notation of our third-order viewpoint; First the language and formula classes of bounded arithmetic, followed by our calculus of third-order functions.

2.1 Bounded Arithmetic

We consider a three-sorted ("third-order") predicate calculus with free and bound variables of the first sort named $a, b, c, ...$ and $x, y, z, ...$, respectively, and free and bound variables of the second sort named $A, B, C, ...$ and $X, Y, Z, ...$, and likewise of the third sort named $\mathcal{A}, \mathcal{B}, \mathcal{C}, ...$ and $\mathcal{X}, \mathcal{Y}, \mathcal{Z}, ...$. The first sort are intended to represent natural numbers; the second, finite sets of natural numbers; and the third, sets (not necessarily finite) of finite binary strings, referred to as "superstrings". The language $\mathcal{L}_A^3$consists of the following set of nonlogical symbols $\mathcal{L}_A^3 = \{0, 1, +, \cdot, |\cdot|, \in_2, \in_3, \leq, =_1, =_2\}$, the same as the set $\mathcal{L}_A^2$for V^1 but with the addition of the third-order membership predicate $A \in_3 \mathcal{B}$. We will often omit the subscripts on '=' and '$\in$' as there is no danger of confusion. Note in particular the absence of the smash function symbol and third-order equality, as well as the length function being present for strings only: the expression $|X|$ is intended to represent 1 plus the largest element of the set X, or 0 if empty. Such sets may be thought of interchangeably with finite binary strings under the following mapping, as in [3]: The set X represents the string with length $|X| - 1$ whose i^{th} bit is 1 exactly when $i \in_2 X$. This map is a bijection with the exception that the string corresponding to the empty set would be undefined, so we define it to be the empty string.

Superstrings (or initial segments thereof) will also sometimes be thought of as finite or infinite strings of bits. The (ordinary) strings indexing the superstring are referred to as **bit-indices**. Since there is no length-function analogue for superstrings, the desired "length" (i.e., lexicographically maximal bit-index under consideration) will have to be specified separately.

Number terms are defined identically as in V^1, in particular not including any reference to third-order variables. Formulas are defined as usual, with the addition of the third-order variables and quantifiers on those variables. There is a hierarchy of classes $\Sigma_i^{\mathcal{B}}$ and $\Pi_i^{\mathcal{B}}$ of formulas in this language analogous to the hierarchies Σ_i^B and Π_i^B of second-order formulas: $\Sigma_i^{\mathcal{B}}$ consists of those formulas with arbitrarily many bounded first- and second-order quantifiers, and exactly i alternations of third-order quantifiers, the outer-most being **restricted**, i.e. equivalent to an existential quantifier. We shall be concerned only with $i \in \{0, 1\}$. In contrast to this more general class, recall the definition of **strict** $\Sigma_1^{\mathcal{B}}$-formulas as those consisting of a single existential third-order quantifier followed by a formula with no third-order quantifiers; we shall also refer to a slightly more inclusive class of formulas called strict $\forall^2\Sigma_1^{\mathcal{B}}$, consisting of a single bounded universal second-order quantifier followed by a strict $\Sigma_1^{\mathcal{B}}$-formula. We shall refer to this class simply as $\forall^2\Sigma_1^{\mathcal{B}}$, omitting the explicit epithet "strict".

Note that third-order quantifiers are not bounded, and in fact there is no apparent way to bound them (short of using an unbounded quantifier of a lower order) due to the lack of a length function. Fortunately, in the appropriate fragments of the theories we shall be concerned with, these variables will always be implicitly bounded, in the sense that the bounds on lower-order quantifiers will limit what part of the superstring actually affects the truth-value of a given formula..

2.2 Third-Order Computation and Function Calculus

In this section we introduce our framework of third-order computation. This includes a calculus of third-order functions and multi-functions that will be used later to obtain universal versions of some theories. The intent is to capture the nature of computation defined by third-order theories of bounded arithmetic. For this reason, we are primarily interested in classes of polynomially-bounded functions (from strings to strings) or similar, as this makes operations such as composition of functions more natural. We are consequently interested in our classes of functions somehow maintaining an exponential-size distinction between the three sorts, as do (standard) theories of bounded arithmetic.

Notwithstanding the above, we make the following definitions as general as possible:

Let $\mathbb{D} = \langle \mathbb{N}, \{0, 1\}^*, 2^{\{0,1\}^*} \rangle$. This will be the three-sorted domain of our multi-functions: The natural numbers, finite binary strings, and **finite** sets of finite binary strings. (**XXX: Notation for finite subsets?**) Note that these sorts are similar to those described earlier for third-order theories of bounded arithmetic, except that the second type is binary strings, not finite sets of numbers, and that the third type are now finite sets of strings. We will use the same typographical conventions as for variables above to refer to members of these domains. Function symbols in our calculi will similarly be named from the lists $f, g, ...; F, G, ...;$ and $\mathcal{F}, \mathcal{G}, ...$ to indicate the sort of the range of the function. Functions or objects of unspecified sort will be named with a tilde such as $\tilde{a}$ or $\tilde{f}$. We will also consider third-order predicates, which for simplicity we will consider as 0-1 valued functions.

Definition 2.1. *Let $\mathbb{E} = \mathbb{E}^1 \cup \mathbb{E}^2 \cup \mathbb{E}^3$ be the set of functions of the three sorts. This includes both functions and multi-functions. The 0-1 valued functions are referred to as $\mathbb{E}^0 \subset \mathbb{E}^1$.*

2.2.1 Computation of Functions

Now, expanding the usual definition of Turing machines computing multi-functions from binary stings to binary strings, we define what it means for Turing machines to compute multi-functions from $\mathbb{E}$:

Definition 2.2. *Let $\mathbb{N}^j \times (\{0,1\}^*)^k \times (2^{\{0,1\}^*})^l$ be the domain of some multi-function in $\tilde{f} \in \mathbb{E}$. Then an oracle machine M* **computes** *the function if for every value of the parameters,*

$$M^{\mathcal{A}_1,...\mathcal{A}_l}(1^{a_1}\#...\#1^{a_j}\#A_1\#...\#A_k)$$

outputs some image of the function onto a special write-only output tape, in unary in the case of a number-valued function.

Not to be confused with the above, if the multi-function is $\mathcal{F} \in \mathbb{E}^3$, we say that M **accepts** *$\mathcal{F}$ if there is some image $\mathcal{W} = \mathcal{F}(a_1, ..., a_j, A_1, ..., A_k, \mathcal{A}_1, ..., \mathcal{A}_l)$ and*

$$M^{\mathcal{A}_1,...\mathcal{A}_l}(1^{a_1}\#...\#1^{a_j}\#A_1\#...\#A_k\#X)$$

accepts (or outputs 1) exactly when $X \in \mathcal{W}$.

We can extend this definition to other computation models operating on strings. The following definition is vague and intended only to give an approximate naming convention. Specific instances of this definition will need clarification.

Definition 2.3 (Meta-Definition). *Let FC be a complexity class of string multi-functions with a well-understood semantics for oracle access such as a query tape, oracle gate, or similar. Then FC^+ denotes the class of multi-functions from $\mathbb{E}$ computed by machines of the model of FC in the sense of the above definition.*

Alternatively, let C be a complexity class of languages, again with an oracle semantics. Then C° denotes the class of multi-functions $\mathcal{F} \in \mathbb{E}^3$ for which there is a machine $M_\mathcal{F}$ of the model of C accepting $\mathcal{F}$ in the sense of the above definition. In this case we ordinarily intend for the resource bounds of the computation to be determined by the arguments to the function only, and not the special query argument X.

Finally, for C a class of languages, $C^\diamond$ denotes the class of 0-1 valued functions $f \in \mathbb{E}^0$ such that there is a machine M_f from the class of C accepts exactly when the value of f is 1.

Some comments about superstring-valued functions are in order: For the purposes of our function calculus, the superstring values of functions are finite, bounded strings, and the (recursion-theoretic) operations defined below will allow only such “well-behaved” superstring-valued functions to be defined. However, the previous definition for C° seems to allow the superstring value of a function to be infinite (for example, the machine accepts every X). In this case, the value of the "function" is undefined, exactly as it would be if the machine ran forever. For the complexity classes of interest to us, it will be trivial to eliminate these problems syntactically (with timers and input length checks).

Note that for some C and FC being closely related classes, for example P and FP, the superstring-valued functions in C° are in general very different from those in FC^+, as the superstring accepted in the former case may be much longer (by an exponential factor) than the one computed in the latter. Thus we must be clear about which paradigm of superstring–valued computation we mean in each circumstance. With these points in mind, call a functions from the calculus **polynomially bounded** if the number outputs, lengths of string outputs and lengths of bit-indices in superstring outputs are all bounded by polynomials in the number inputs and lengths of string inputs of the function. We deliberately exclude superstring inputs from the computation of this bound.

$\mathrm{FP}^+ \cup \mathrm{P}^\circ$ is then the class of polynomially bounded polytime functions. $\mathrm{FPSPACE}^+$ is similarly the class of polynomially bounded PSPACE functions (where the space bound applies to both the work tape and all query tapes used for reading superstring inputs). FEXP^+ is the class of polynomially bounded exponential time functions, where the space used on query tapes is also polynomially bounded. Note that unlike the case for P, $\mathrm{PSPACE}^\circ \subset \mathrm{FPSPACE}^+$, and $\mathrm{EXP}^\circ \subset \mathrm{FEXP}^+$. Restricted to functions from strings to strings, these classes are just the usual function classes.

Some classes of third-order predicates, our analogue of languages, are the following: $\mathrm{P}^\diamond$, $\mathrm{NP}^\diamond$, $\mathrm{PSPACE}^\diamond$, $\mathrm{NPSPACE}^\diamond$, $\mathrm{EXP}^\diamond$ and $\mathrm{NEXP}^\diamond$ are the predicates computable in polynomial time, nondeterministic polynomial time, polynomial and nondeterministic polynomial space, and exponential and nondeterministic exponential time, respectively. Restricted to unary predicates on strings, these would be exactly the classes of characteristic functions of the usual complexity classes of languages. It is easy to see, however, that $\mathrm{P}^\diamond \neq \mathrm{NP}^\diamond$, as a predicate in the latter class can determine if a given superstring contains a 1 (up to a bound given by a string argument), while this predicate is clearly not in $\mathrm{P}^\diamond$. It does seem that the usual argument for Savitch's theorem goes through, at least for $\mathrm{NPSPACE}^\diamond$: configurations are still described by polynomial-sized strings. We conclude that $\mathrm{PSPACE}^\diamond = \mathrm{NPSPACE}^\diamond$.

XXX: Reference Klote-Takeuti for higher-order oracle semantics.

Now, in order to expand our discussion to the exponential-time hierarchy, we must first address relativising classes of functions by adding oracles. It is most fervently desired that the reader not confuse these third-order oracles (our generalization of ordinary oracles) with the above use of oracle machines to receive third-order inputs.

We now define our third-order version of oracle relativisation:

Definition 2.4. *A* **third-order oracle** *Turing machine has a number of specified write-only query tapes, each one designated with a sort. The machine may write polynomially-bounded values on these tapes, in the sense that the numbers (in unary), lengths of strings, and bit-indices of superstrings written are all bounded by fixed polynomials in the machine's (non-superstring) inputs. When the machine enters the special query state, these tapes are erased, and a value is returned to the machine by way of a special read-only reply tape (with random access in the case of a superstring-valued oracle).*

Now paralleling the definition of the ordinary complexity classes $\Sigma_i^{exp} (= \mathrm{NEXP}^{\Sigma_{i-1}^p})$ of the nondeterministic exponential-time hierarchy we can define the corresponding classes of 0-1 valued functions from $\mathbb{E}^1$. It is important to observe that the queries made of the Σ_{i-1}^p oracle by the NEXP machine in the standard definition are in general of exponential size. We define $(\Sigma_1^{exp})^\diamond = \mathrm{NEXP}^\diamond$ and $(\Sigma_i^{exp})^\diamond = (\mathrm{NEXP}^\diamond)^{(\Sigma_{i-1}^{exp})^\diamond}$. In other words, each higher level of the hierarchy is obtained by augmenting nondeterministic exponential time with a third-order oracle for the previous level. Since the queries to this oracle must be polynomially bounded per our definition, it can be seen that this relativisation corresponds to unbounded access to an oracle from the appropriate level of the quasi-polynomial-time hierarchy (considered as a predicate on the superstring inputs); however in the hands of an NEXP machine such an oracle is no more powerful than one from the same level of the polynomial-time hierarchy, as the machine could simply make polynomially longer queries of this latter oracle. Thus as predicates purely on strings, the levels of our hierarchy correspond precisely with the levels of the ordinary exponential-time hierarchy.

The function classes $(\square_i^{exp})^+$=(FEXP$^{(\Sigma_{i-1}^{exp})^\diamond}$)$^+$ are the classes of polynomially bounded functions computed by exponential-time Turing machines relativized with a third-order oracle for a function from $(\Sigma_i^{exp})^\diamond$, and similarly as functions purely of strings correspond to the usual $\square_i^{exp}$.

It should be noted that $(\Sigma_i^{exp})^\diamond = (\Pi_i^{exp})^\diamond$ seems to imply that the third-order exponential-time hierarchy collapses to the ith level, while this is not known for the ordinary case.

2.2.2 Recursion Theory of Functions

Now let us define some standard functions. The number functions $\{x+y, x{\cdot}y\}$, constants 0,1, etc. are as usual. The bit, string successor and concatenation functions $\{\text{bit}(x,Y), s_0(X), s_1(X), X \frown Y\}$ are also standard. $\{|X|, X \in \mathcal{Y}, 1^x\}$ respectively give the length of a string, the (0-1-valued) characteristic function of $\mathcal{Y}$, and a standard string of length x.

We now define several operations on these functions. As our focus is on string functions as opposed to the standard recursion-theoretic viewpoint of number functions, we will comment in each case on how these operations compare to standard operations on number functions.

Define $\tilde{f}$ (of any sort) by **limited recursion** from $\tilde{g}$, $\tilde{h}$ (also of any sort) and l by $\tilde{f}(0,...) = \tilde{g}(...), \tilde{f}(x+1,...) = \tilde{h}(x, \tilde{f}(x,...),...)$ and either $\tilde{f}(x,...) \le l(x,...)$ or $|\tilde{f}(x,...)| \le l(x,...)$, as appropriate. This operation corresponds roughly to limited recursion on notation for number functions, as it iterates a function ($\tilde{h}$) a polynomial number of times subject to a bound on growth. **Recursion** is the same operation without the bound on growth.

Define $\tilde{f}$ by **limited doubling recursion** from $\tilde{g}$ and l by $\tilde{f}(0,\tilde{y},...) = \tilde{g}(\tilde{y},...), \tilde{f}(x+1,\tilde{y},...) = \tilde{f}(x, \tilde{f}(x,\tilde{y},...),...)$ and either $\tilde{f}(x,\tilde{y},...) \le l(x,...)$ or $|\tilde{f}(x,\tilde{y},...)| \le l(x,...)$, as appropriate. This operation corresponds roughly to limited recursion for number functions, as it iterates a function ($\tilde{g}$) an exponential number of times (by doubling the number of nestings a polynomial number of times) subject to a bound on growth. **Doubling recursion** is the same operation without the bound on growth.

Define $\tilde{f}$ (of any sort) by **limited long recursion** from $\tilde{g}$, $\tilde{h}$ (also of any sort) and l by $\tilde{f}(1^0,...) = \tilde{g}(...), \tilde{f}(X+1,...) = \tilde{h}(x, \tilde{f}(X,...),...)$ and either $\tilde{f}(X,...) \le l(X,...)$ or $|\tilde{f}(X,...)| \le l(X,...)$, as appropriate. This operation is similar to the previous one in that it iterates a function an exponential number of times; however, it differs in that the exponentially many iterations are performed directly by using a string as an exponential-length counter. This operation presupposes a suitable string successor function $X+1$.

Define $\mathcal{F}$ by **limited 3-comprehension** from $g, h \in \mathbb{E}^1$ by $\mathcal{F}(..)(X) \leftrightarrow (|X| \le g(..) \wedge h(X,..) = 0)$.

Observe that each of the above operations, when used to obtain a superstring-valued function, in fact defines a unique function. For our purposes pertaining to theories of bounded arithmetic (in which superstrings are not bounded), it is important to distinguish these "pure" operations from bounded versions of them that only specify an initial segment of the superstring value of the function. The computational object so defined is now a multifunction, as there are many correct images of the multifunction for any set of parameters. For example, here is a bounded version of $\mathcal{F}$ defined by limited 3-comprehension from g and h: $|X| \leq g(...) \supset (\mathcal{F}(..)(X) \leftrightarrow h(X, ..) = 0)$.

It should be noted here that the recursion operations, as well as simple composition of functions, appear to be significantly more powerful when applied to superstring-valued functions. This is because in the composition of two such functions, the space may not be available to write down the intermediate value. A space-bounded computation model would then have to query the "inner" function many times (to retrieve bits of its output as needed) in order to compute the outer function. The composition of two polynomially bounded number- or string-valued functions appears to require the sum of the time requirements (computing first one then the other function), while the required space does not increase. For superstring-valued functions, on the other hand, the time required for the composition as described seems in general to be the product of the time required for each component, while the space required is the sum. If space is not bounded then the intermediate results can be written in full, and thus time and space requirements are as for the composition of number- or string-valued functions.

At this point we can extrapolate a bit from Cobham to see that $\mathrm{FP}^{+} \cup \mathrm{P}^{\circ}$ is exactly the closure of $I = \{0, 1, x + y, x \cdot y, 1^x, |X|, s_0(X), s_1(X), \mathrm{bit}(x, Y), X \frown Y, X \in \mathcal{Y}\}$ under composition, limited 3-comprehension and limited recursion with the latter restricted to $\mathbb{E}^1 \cup \mathbb{E}^2$. Probably fewer initial functions suffice but these are all in FP^{+} and are enough to generate the entire class.

Proof Sketch. With $\{0, 1, x + y, x \cdot y, 1^x, |X|\}$ we can construct numbers of polynomial magnitude (as a function of the lengths of string inputs) and thus also strings of polynomial length. Now we have essentially all the initial functions of Cobham. Limited recursion together with bit can then simulate limited recursion on notation. Once all functions from $\mathbb{E}^2 \cap \mathrm{FP}^{+}$ are available, the rest of the class is easily obtained.

All these initial functions are clearly in $\mathrm{FP}^{+} \cup \mathrm{P}^{\circ}$, which is closed under limited recursion (not on $\mathbb{E}^3$), limited 3-comprehension and composition. □

To obtain FP^{+} but not P°, we would need an alternative way to obtain superstrings, as in this case 3-comprehension produces exponentially longer superstrings than functions from FP^{+} can output. As this is not our focus here, we leave this problem as an exercise. **XXX: Can I say that in a thesis?**

$\mathrm{FPSPACE}^{+}$ is contained in the closure of $\mathrm{FP}^{+} \cup \mathrm{P}^{\circ}$ by limited recursion on $\mathbb{E}^3$, composition and limited 3-comprehension: First, a superstring-valued $\mathrm{FP}^{+} \cup \mathrm{P}^{\circ}$ function can compute from the input of a PSPACE Turing machine the transition function of the machine as a table listing the next configuration for each given configuration. Another function in $\mathrm{FP}^{+} \cup \mathrm{P}^{\circ}$ can compose such a function with itself by reading two (polynomial-sized) entries from this table. Therefore after applying limited recursion on these two functions we obtain a third that outputs the 2^x-step transition function and from this it is trivial to extract the value of the original PSPACE function. Since $\mathrm{FPSPACE}^{+}$ is closed under limited recursion (as each such operation increases the space requirements of a function by a polynomial factor), limited 3-comprehension and composition, we can conclude that this class is exactly the closure of the initial functions under these operations.

$\mathrm{FPSPACE}^{+}$ is alternatively characterized as the closure of $\mathrm{FP}^{+} \cup \mathrm{P}^{\circ}$ under composition and limited doubling recursion restricted to $\mathbb{E}^1 \cup \mathbb{E}^2$. The step function of a PSPACE Turing machine can be iterated exponentially many times using these operations, and conversely $\mathrm{FPSPACE}^{+}$ is closed under limited doubling recursion as the recursion can be unwound with only a polynomial amount of additional space.

FEXP^{+} is the closure of $\mathrm{FP}^{+} \cup \mathrm{P}^{\circ}$ under limited doubling recursion on $\mathbb{E}^3$: The step function of an exponential-time Turing machine can be iterated exponentially many times.

3 Some Third-order Theories

In this section we give the definition of several third-order theories and axiom schemes.

W_1^i is a theory over $\mathcal{L}_A^3$. The axioms of W_1^i are B1-B12, L1, L2 and SE of [Cook/Kolokolova], (strict) $\forall^2\Sigma_i^{\mathcal{B}}$-IND and the following two comprehension schemes $\Sigma_0^{\mathcal{B}}$-2COMP:

$$(\exists Y \le t(\overline{x}, \overline{X}))(\forall z \le a)[\phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, z) \leftrightarrow Y(z)]$$

and $\Sigma_0^{\mathcal{B}}$-3COMP:

$$(\exists \mathcal{Y})(\forall Z \le a)[\phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, Z) \leftrightarrow \mathcal{Y}(Z)],$$

where in each case $\phi \in \Sigma_0^{\mathcal{B}}$ subject to the restriction that neither Y nor $\mathcal{Y}$, as appropriate, occurs free in ϕ.

W_1^1 defined above is slightly different than the version published in CSL04 [7]; it includes a string equality symbol and extensionality axiom. This predicate is $\Delta_0^{\mathcal{B}}$-definable in the original version of the theory and can thus be conservatively added and used in all the axiom schemes.

Define $\widehat{W_1^i}$ to be the analogous theory with the induction scheme restricted to **strict** $\Sigma_i^{\mathcal{B}}$ formulas. Note that $\widehat{W_1^0} = W_1^0$.

TW_1^i is defined identically as above, but with the following scheme named $\Sigma_i^{\mathcal{B}}$-SIND (string induction) in place of $\Sigma_i^{\mathcal{B}}$-IND:

$$[\forall X, Y, Z((|Z| = 0 \supset \phi(Z)) \wedge (\phi(X) \wedge S(X, Y) \supset \phi(Y)))] \supset \forall Z \phi(Z)$$

for $\phi \in$(strict)$\Sigma_i^{\mathcal{B}}$, where $S(X, Y)$ is the following formula expressing that Y is the lexicographically next finite set after X:

$$|Y| \le |X| + 1 \wedge \exists i \le |Y|[Y(i) \wedge \neg X(i) \wedge \forall j < i(X(j) \wedge \neg Y(j)) \wedge \forall j \le |Y|(i < j \supset (X(j) \leftrightarrow Y(j)))].$$

(This formulation is due to Phuong Nguyen).

TTW_1^i is yet another theory in this vein, with a yet stronger induction scheme named $\Sigma_i^{\mathcal{B}}$-SSIND ("superstring" induction). Note that since (by design) there is no way to bound a third-order object, the scheme refers to a term t, and restricts its attention to the first 2^t bits of the objects. It is intended that this t be some crucial bound from ϕ. The scheme is:

$$[\forall \mathcal{X}, \mathcal{Y}, \mathcal{Z}((\forall X \le t \neg \mathcal{Z}(X)) \supset \phi(\mathcal{Z})) \wedge (\phi(\mathcal{X}) \wedge S_3(\mathcal{X}, \mathcal{Y}, t) \supset \phi(\mathcal{Y}))] \supset \forall \mathcal{Z} \phi(\mathcal{Z})$$

for $\phi \in$(strict)$\Sigma_i^{\mathcal{B}}$, where $S_3(\mathcal{X}, \mathcal{Y}, z)$ is the formula

$$\begin{aligned} &\exists Z \le z[\mathcal{Y}(Z) \wedge \neg \mathcal{X}(Z) \wedge \\ &\qquad \forall X \le z(L_2(X, Z) \supset \mathcal{X}(X) \wedge \neg \mathcal{Y}(X)) \wedge \forall X \le z(L_2(Z, X) \supset \mathcal{X}(X) \leftrightarrow \mathcal{Y}(X))) \end{aligned}$$

and $L_2(X, Y)$ is the formula

$$\exists i \le |X|[Y(i) \wedge \neg X(i) \wedge \forall j \le |Y|(i < j \supset (X(i) \leftrightarrow Y(i)))].$$

The scheme $\Sigma_0^{\mathcal{B}}$-superstring-recursion is the following:

$$\exists \mathcal{X} \phi^{\text{rec}}(x, \mathcal{X}),$$

where $\phi(Y, \mathcal{X}) \in \Sigma_0^{\mathcal{B}}$, and

$$\phi^{\text{rec}}(x, \mathcal{X}) \equiv \forall Y \le x(\mathcal{X}(Y) \leftrightarrow \phi(Y, \mathcal{X}^{<Y})),$$

where $\mathcal{X}^{<Y}$ is a chop function and x is not free in ϕ. ϕ (and therefore also ϕ^{rec}) may have other free variables than the displayed ones, but ϕ must have distinguished string and superstring free variables Y and $\mathcal{X}$. ϕ^{rec} then has $\mathcal{X}$ free as well as a new variable x.

The scheme $\Sigma_0^{\mathcal{B}}$-superstring-halfrecursion is the following:

$$\exists \mathcal{X} \phi^{\mathrm{hrc}}(x, \mathcal{X}),$$

where $\phi(Y, \mathcal{X}) \in \Sigma_0^{\mathcal{B}}$, and

$$\phi^{\mathrm{hrc}}(S, \mathcal{X}) \equiv \forall Y \leq x(\mathcal{X}(Y) \leftrightarrow \phi(Y, \mathcal{X}^{<Y/2})),$$

where $\mathcal{X}^{<Y/2}$ is a chop function returning the first $\frac{Y}{2}$ (as a number) bits of $\mathcal{X}$. ϕ and ϕ^{rec} have the same free-variable conventions and requirements as in the superstring recursion scheme.

Then HW_1^0 is the theory W_1^0 with the addition of the $\Sigma_0^{\mathcal{B}}$-superstring-halfrecursion scheme.

4 Third-Order Parikh's Theorems

In this section we prove a generalization of Parikh's theorem for third-order theories. First, some definitions:

Definition 4.1. *A formula is* **2-bounded** *if all of its first- and second-order quantifiers are bounded. (It may contain arbitrary third-order quantifiers).*

Let T be a theory extending W_1^0 and $\mathcal{L} \supseteq \mathcal{L}_A^3$ be the vocabulary of T. Then T is a **2-bounded theory** *if it is axiomatized by 2-bounded formulas.*

Definition 4.2. *Let $M = \langle M_1, M_2, M_3 \rangle$ be a model of B1-B12, L1, L2, SE. Analogously to the first-order case, a 2-cut in M is any subset $I = \langle I_1 \subseteq M_1, I_2 \subseteq M_2, I_3 = M_3 \rangle$ closed under $x + 1$ and $\leq$ (for numbers and strings). This last point means that if $b \in I_1$ and $M \models a \leq b$ for $a \in M_1$ (or $M \models |A| \leq b$ for $A \in M_2$), then $a \in I_1$ (respectively, $A \in I_2$). For a string $A \in M_2$, it is equivalent to say that if $|A| \in I_1$ then $A \in I_2$.*

This is denoted $I \subseteq_e^2 M$.

Lemma 4.3. *Let M be a third-order structure with vocabulary $\mathcal{L}$ and and $I \subseteq_e^2 M$ be a 2-cut of M closed under all the function symbols in $\mathcal{L}$. Finally, let $\phi(\overline{a}, \overline{A}, \overline{\mathcal{A}})$ be a 2-bounded formula with all free variables displayed, and $\overline{b}, \overline{B}, \overline{\mathcal{B}} \in I$. Then*

$$I \models \phi(\overline{b}, \overline{B}, \overline{\mathcal{B}}) \qquad \textit{iff} \qquad M \models \phi(\overline{b}, \overline{B}, \overline{\mathcal{B}}).$$

Proof Sketch. This lemma is proved by induction on the quantifier complexity of ϕ.

The base case is quantifier-free formulas, and is clear, as all parameters are in the cut.

For the induction step, consider $I \models \forall X \leq t\phi(X)$. All parameters (including those in t), not shown, are from I. Then $I \models \phi(B)$ for each $B \leq t$. But then $M \models \phi(B)$ for each $B \leq t$ in M as all such elements are already in I, and so $M \models \forall X \leq t\phi(X)$. The other direction ($M \models \forall X \leq t\phi(X) \implies I \models \forall X \leq t\phi(X)$) is easier as the range of the universal quantifier is decreased.

The case of a first-order quantifier is very similar, and existential quantifiers are handled symmetrically. The case of a third-order quantifier is straightforward, as the range of the quantifier remains the same in M or I. □

Note that the above lemma does not require any assumption about function symbols being bounded by monotone terms. It also does not restrict the sorts of the range or any component of the domain of a function symbol.

At this point it is apparent that the open axioms B1-B12, L1 and L2 as well as SE and the comprehension and recursion schemes, are satisfied in any $\mathcal{L}$-closed 2-cut of any model of any 2-bounded theory T, as all are 2-bounded. In fact, the same is true of the induction schemes, even the sharply bounded ones. This is because they all have 2-bounded versions. The 2-bounded B-$\Sigma_i^{\mathcal{B}}$-SIND is:

$$[\forall Z \leq 0 \forall X \leq |W| \forall Y \leq |W| (\phi(Z) \wedge (\phi(X) \wedge S(X,Y) \supset \phi(Y)))] \supset \phi(W)$$

and the 2-bounded B-$\Sigma_i^{\mathcal{B}}$-IND is:

$$[\phi(0) \wedge \forall x \leq w(\phi(x) \supset \phi(x+1))] \supset \phi(w).$$

These bounded induction schemes logically imply the unbounded versions. Conversely, the unbounded schemes prove the bounded ones: for example, $\Sigma_i^{\mathcal{B}}$-SIND on the formula $|X| \leq |W| \supset \phi(X)$ gives

$$[\forall X, Y, Z((|Z| = 0 \supset (|Z| \leq |W| \supset \phi(Z))) \wedge ((|X| \leq |W| \supset \phi(X)) \wedge S(X,Y) \supset (|Y| \leq |W| \supset \phi(Y))))] \supset \forall Z(|Z| \leq |W| \supset \phi(Z)).$$

By strengthening the hypothesis,

$$[\forall Z \leq 0 \forall X, Y(\phi(Z) \wedge ((|X| \leq |W| \supset \phi(X)) \wedge S(X,Y) \supset (|Y| \leq |W| \supset \phi(Y))))] \supset \forall Z(|Z| \leq |W| \supset \phi(Z))$$

and again by the fact that if $S(X,Y)$ then $|X| \leq |Y|$,

$$[\forall Z \leq 0 \forall X \leq |W| \forall Y \leq |W| (\phi(Z) \wedge (\phi(X) \wedge S(X,Y) \supset \phi(Y)))] \supset \forall Z(|Z| \leq |W| \supset \phi(Z)).$$

Finally,

$$[\forall Z \leq 0 \forall X \leq |W| \forall Y \leq |W| (\phi(Z) \wedge (\phi(X) \wedge S(X,Y) \supset \phi(Y)))] \supset \phi(W),$$

which is B-ϕ-SIND.

Thus we have proven that all the theories described above in section 3 are in fact 2-bounded.

A corollary of the previous lemma:

Corollary 4.4. *Let T be any 2-bounded extension of W_1^0 M be a model of T. Let $I \subseteq_e^2 M$ be a 2-cut of M closed under all the function symbols in $\mathcal{L}$. Then the (2-bounded) axioms of T are satisfied by I, and consequently, $I \models T$.*

We require one additional definition before stating Parikh's theorem for first- and second-order existential quantifiers:

Definition 4.5. *Let T be a three-sorted theory with vocabulary $\mathcal{L} \supseteq \mathcal{L}_A^3$ containing the open axioms of W_1^0. We say that T has* **monotone 2-bounding** *if the following hold:*

1. *For every number-valued function symbol $f \in \mathcal{L}$, there is a number term t_f of $\mathcal{L}$ such that*

$$T \vdash \overline{\tilde{a}} \leq \overline{b} \supset f(\overline{\tilde{a}}) \leq t_f(\overline{b}),$$

where $\overline{\tilde{a}}$ is a list of variables of any order and $\overline{b}$ is a list of number variables, and $\overline{\tilde{a}} \leq \overline{b}$ abbreviates a conjunction of subformulas of the form $a_i \leq b_i$ or $|A_i| \leq b_i$, as appropriate, for each $\tilde{a}_i$ not a third-order variable.

2. *For every string-valued function symbol $F \in \mathcal{L}$, there is a term t_F of $\mathcal{L}$ such that $T \vdash \overline{\tilde{a}} \leq \bar{b} \supset |f(\overline{\tilde{a}})| \leq t_f(\bar{b})$.*

Theorem 4.6 (Third-Order Parikh's Theorem). *Let $\phi(\overline{\tilde{x}})$ be a 2-bounded formula, all free variables displayed, and T a 2-bounded extension of W_1^0 with vocabulary $\mathcal{L}$ and monotone 2-bounding.*

Further, assume that $T \vdash \forall \overline{\tilde{x}} \exists \tilde{y} \phi(\overline{\tilde{x}}, \tilde{y})$, where $\overline{\tilde{x}}$ are of any sort and $\tilde{y}$ is first- or second-order. Then there is some term t such that $T \vdash \forall \overline{\tilde{x}} \exists \tilde{y} \leq t(\overline{\tilde{x}}) \phi(\overline{\tilde{x}}, \tilde{y})$.

Proof. This theorem is proved by a compactness argument. Assume the hypothesis of the theorem, and furthermore that $T \nvdash \forall \overline{\tilde{x}} \exists \tilde{y} \leq t(\overline{\tilde{x}}) \phi(\overline{\tilde{x}}, \tilde{y})$ for any term t. Then by compactness the theory

$$T' = T + \{\forall \tilde{y} \leq t(\overline{\tilde{c}}) \neg \phi(\overline{\tilde{c}}, \tilde{y}) : t \text{ any term of } \mathcal{L}\}$$

is consistent with $\overline{\tilde{c}}$ new constants of the appropriate sorts.

Now, let $M \models T'$ and define $I \subseteq_e^2 M$ by $b \in I_1$ (respectively, $B \in I_2$) iff there is a term t such that $M \models b < t(\overline{\tilde{c}})$ ($M \models |B| < t(\overline{\tilde{c}})$). (And $I_3 = M_3$). It is evident that I is indeed a 2-cut of M, but for I to be $\mathcal{L}$-closed, it is essential at this point for T to have monotone 2-bounding. Otherwise, there could be some $b \in I$ by virtue of $M \models b \leq t(\overline{\tilde{c}})$, yet $f(b) \notin I$ for f is not monotone. However, our assumption ensures that $M \models f(b) \leq t_f(t(\overline{\tilde{c}}))$. The monotone 2-bounding assumption implies that applying function symbols to elements in I (which are bounded by terms in $\overline{\tilde{c}}$) produces elements which are also bounded, and so already in I.

I is then a model for T by the corollary, and yet $I \models \exists \overline{\tilde{x}} \forall \tilde{y} \neg \phi(\overline{\tilde{x}}, \tilde{y})$, a contradiction. □

5 Facts, Conjecture and Questions About the Theories

In this section we summarize what is known about the various theories described above. Many of these facts are proved in subsequent sections on definability or witnessing theorems. The terminology and notation is admittedly loose in this section as we are chronicling work in progress. Please see the following sections for the definitions of definability of third-order functions.

Lemma 5.1. *W_1^0 is a conservative extension of V, the two-sorted theory for the polytime hierarchy.*

Proof outline. The only axioms stating the existence of third-order elements are the comprehension axioms. Any model of V can therefore be expanded to a model of W_1^0 merely by adding third-order elements to satisfy each comprehension instance ($\Sigma_0^{\mathcal{B}}$ formula with parameters from the model under construction). For a general $\Sigma_0^{\mathcal{B}}$-3COMP instance with free variables, we must supply an object satisfying the instance for each set of parameters from the model assigned to the free variables. $\Sigma_0^{\mathcal{B}}$ formulas are closed under substitution of formulas for free third-order variables, so ultimately each comprehension instance unwinds to a $\Sigma_0^{\mathcal{B}}$ formula. In the end, adding these third-order elements will not affect the truth-value of purely second-order formulas. □

The $\Sigma_0^{\mathcal{B}}$-definable functions of W_1^0 are thus FPH$^+ \cup$FPH$^\circ$ (this requires more justification). Following the usual argument, a function symbol for any such function can be added to W_1^0 and results in a conservative extension. The $\Sigma_1^{\mathcal{B}}$-definable functions of W_1^0 are also FPH$^+ \cup$FPH$^\circ$, as the usual witnessing argument for V can be extended to handle third-order existential quantifiers, which arise only because of the comprehension axioms and introduction rules, and in either case are witnessed by FPH$^+ \cup$FPH$^\circ$ functions.

The $\Sigma_1^{\mathcal{B}}$-definable functions of W_1^1 are exactly FPSPACE$^+$ [7].

The $\Sigma_1^{\mathcal{B}}$-definable functions of $\widehat{W_1^1}$ are also exactly FPSPACE$^+$.

A big question is whether or not this theory proves the replacement schemes from the next section.

XXX: WPV, universal third-order polytime theory?

XXX: Question: If $\widehat{W_1^1}$ is conservative over a minimal theory for PSPACE e.g. HW_1^0, does this imply any complexity collapse?

The $\Sigma_1^{\mathcal{B}}$-definable functions of TW_1^0 are also only FPH$^+\cup$FPH$^\circ$. TW_1^0 should be a conservative extension of TV, which is V with the addition of $\Sigma_\infty^{\mathcal{B}}$-SIND, but $TV = V$. In fact,

Lemma 5.2. $W_1^0 = TW_1^0$

Proof. We use the same "shortening of cuts" technique for showing $S_2^{i+1} \supseteq T_2^i$ proves $W_1^0 \vdash \Sigma_0^{\mathcal{B}}$-SIND:

Let $\phi(X) \in \Sigma_0^{\mathcal{B}}$ and A a parameter. Define

$$\psi(x) := \forall X \leq x \forall Y \leq |A| \forall Z \leq |A| (\phi(Y) \wedge \text{Plus}(X, Y, Z) \longrightarrow \phi(Z)).$$

(For a suitable function symbol Plus). Now, trivially $W_0^1 \vdash \psi(0)$. Also,

$$W_1^0 \vdash (\forall X \forall Y(\phi(X) \wedge S(X, Y) \longrightarrow \phi(Y))) \longrightarrow (\psi(x) \longrightarrow \psi(x+1))$$

by considering the two cases of the low-order bit of X (in $\psi(x+1)$) and applying the induction step of $\Sigma_0^{\mathcal{B}}$-SIND if necessary. Thus by $\Sigma_0^{\mathcal{B}}$-IND, $W_1^0 \vdash \psi(|A|)$. This and the remaining hypothesis of $\Sigma_0^{\mathcal{B}}$-SIND, $\forall X(|X| = 0 \longrightarrow \phi(X))$, imply $\phi(A)$. Therefore $W_1^0 \vdash \Sigma_0^{\mathcal{B}}$-SIND and thus $W_1^0 = TW_1^0$. □

The $\Sigma_1^{\mathcal{B}}$-definable functions of TTW_1^0 are EXP. This is because TTW_1^0 proves $\Sigma_0^{\mathcal{B}}$-superstring-recursion:

$$\exists \mathcal{X} \phi^{\text{rec}}(Y, \mathcal{X}),$$

where $\phi \in \Sigma_0^{\mathcal{B}}$, and

$$\phi^{\text{rec}}(S, \mathcal{X}) \equiv \forall Y \leq |S| (\mathcal{X}(Y) \leftrightarrow \phi(Y, \mathcal{X}^{<Y})),$$

where $\mathcal{X}^{<Y}$ is a chop function. See section 7 for details.

The $\Sigma_1^{\mathcal{B}}$-definable functions of HW_1^0 are exactly FPSPACE$^+$.

6 $\Sigma_1^{\mathcal{B}}$-Replacement Schemes

In this section we shall show that various replacement schemes, allowing third-order existential quantifiers to be moved past lower-order quantifiers, are theorems of W_1^1.

First, though, it is convenient to note that adding to W_1^1 function symbols for its number- and string-valued $\Sigma_1^{\mathcal{B}}$-definable functions results in a conservative extension. The proof of the present claim is analogous to that for first-order bounded arithmetic theories in section 2.3 of [1]. In that proof, a given Σ_1^b-formula in the augmented language is shown to be equivalent to a constructed Σ_1^b-formula in the original language.

$W_1^1 \supset V (= \bigcup V^i)$ since all the axioms of the latter theory are in the former. W_1^1 can therefore $\Sigma_0^{\mathcal{B}}$-define all number- and string-valued functions of number and string arguments from the polynomial-time hierarchy. By the remarks in the previous paragraph, we can add symbols for these functions to W_1^1 and obtain a conservative extension. In particular, pairing functions such as $< x, y >$, $< X, Y >$ and $< X, y >$ may be added. For a third-order variable $\mathcal{X}$ define $\mathcal{X}^{[x]}(X) \equiv \mathcal{X}(< x, X >)$ and $\mathcal{X}^{[X]}(Y) \equiv \mathcal{X}(< X, Y >)$, which make $\mathcal{X}$ into an array, with rows indexed by number or strings respectively, each row of which is a third-order object. With this in mind, we can state the $\Sigma_1^{\mathcal{B}}$ replacement schemes:

Definition 6.1 ($\Sigma_1^{\mathcal{B}}$ Replacement Schemes). *$\Sigma_1^{\mathcal{B}}$-1REPL is:*

$$\forall x \leq y \exists \mathcal{X} \phi(x, y, \mathcal{X}) \leftrightarrow \exists \mathcal{X} \forall x \leq y \phi(x, y, \mathcal{X}^{[x]})$$

and $\Sigma_1^{\mathcal{B}}$-2REPL is:

$$\forall X \leq y \exists \mathcal{X} \phi(X, y, \mathcal{X}) \leftrightarrow \exists \mathcal{X} \forall X \leq y \phi(X, y, \mathcal{X}^{[X]}),$$

where in each case ϕ is a (general) $\Sigma_1^{\mathcal{B}}$-formula which may have other free variables than those indicated.

Theorem 6.2. *The $\Sigma_1^{\mathcal{B}}$ replacement schemes are theorems of W_1^1.*

Proof. Although the $\Sigma_1^{\mathcal{B}}$-1REPL scheme has a simpler proof, it can also be proved in the same way as the $\Sigma_1^{\mathcal{B}}$-2REPL scheme, so we include only a proof of the latter.

$\leftarrow$: This direction of the equivalence, namely that for $\phi(X, y, \mathcal{X}) \in \Sigma_1^{\mathcal{B}}$

$$W_1^1 \vdash \exists \mathcal{X} \forall X \leq y \phi(X, y, \mathcal{X}^{[X]}) \supset \forall X \leq y \exists \mathcal{X} \phi(X, y, \mathcal{X})$$

is immediate.

$\rightarrow$: The existence of a proof in W_1^1 of this direction of the equivalence is itself proved by structural induction on ϕ. The base case of the induction is when ϕ is $\Sigma_0^{\mathcal{B}}$. Let ψ be $\forall X \leq y \exists \mathcal{X} \phi(X, y, \mathcal{X})$. Let $\theta(c)$ be the formula

$$\forall X \leq (y \dot{-} c) \exists \mathcal{X} \forall Y \leq c \phi(X \frown Y, y, \mathcal{X}^{[Y]}).$$

$\theta(0)$ is a simple logical consequence of ψ, and $W_1^1 \vdash \psi \wedge \theta(c) \supset \theta(c+1)$ by use of $\Sigma_0^{\mathcal{B}}$-3COMP to combine two third-order objects (coding the two arrays of third-order objects for all strings of length smaller than y starting with $X \frown 0$ and $X \frown 1$ respectively) into one third-order object coding the array for all strings of length smaller than y starting with X. Thus $W_1^1 \vdash \psi \supset \theta(y)$ by $\forall^2\Sigma_1^{\mathcal{B}}$-IND, and clearly $W_1^1 \vdash \theta(y) \supset \exists \mathcal{X} \forall X \leq y \phi(X, y, \mathcal{X}^{[X]})$.

Now let $k > 0$ and assume the present theorem holds for every member of $\Sigma_1^{\mathcal{B}}$ with fewer than k third-order quantifiers. Let $\phi \in \Sigma_1^{\mathcal{B}}$ have exactly k third-order quantifiers and assume without loss of generality that ϕ is in prenex normal form. (Every formula is provably in W_1^1 equivalent to one in prenex normal form). Then every third-order quantifier in ϕ is existential, and $\phi(X, y, \mathcal{X})$ is of the form $Q_1\tilde{a_1}...Q_n\tilde{a_n}\exists \mathcal{Z} \psi(\tilde{a_1}, ..., \tilde{a_n}, \mathcal{Z}, X, y, \mathcal{X})$ for some n and ψ with $k-1$ existential third-order quantifiers. Each Q_i is a bounded first- or second-order quantifier and the corresponding $\tilde{a_i}$ is a variable of the appropriate sort. By several applications of the inductive hypothesis we prove

$$Q_1\tilde{a_1}...Q_n\tilde{a_n}\exists \mathcal{Z} \psi(\tilde{a_1}, ..., \tilde{a_n}, \mathcal{Z}, X, y, \mathcal{X}) \supset \exists \mathcal{Z} Q_1\tilde{a_1}...Q_n\tilde{a_n} \psi(\tilde{a_1}, ..., \tilde{a_n}, \mathcal{Z}^{[\tilde{a_1}]...[\tilde{a_n}]}, X, y, \mathcal{X}). \quad (6.1)$$

The inductive hypothesis is not needed for those Q_i which are existential, nor in that case need we add $[\tilde{a_i}]$ to the formula on the right of the equivalence, yet it is harmless and simplifies matters to do so.

Now with $\Sigma_0^{\mathcal{B}}$-3COMP we can prove

$$\exists \mathcal{X} \exists \mathcal{Z} Q_1\tilde{a_1}...Q_n\tilde{a_n} \psi(\tilde{a_1}, ..., \tilde{a_n}, \mathcal{Z}^{[\tilde{a_1}]...[\tilde{a_n}]}, X, y, \mathcal{X}) \supset \\ \exists \mathcal{Z} Q_1\tilde{a_1}...Q_n\tilde{a_n} \psi(\tilde{a_1}, ..., \tilde{a_n}, \mathcal{Z}^{[1][\tilde{a_1}]...[\tilde{a_n}]}, X, y, \mathcal{Z}^{[2]}) \quad (6.2)$$

and thus piecing together implications 6.1 and 6.2 we obtain

$$\forall X \leq y \exists \mathcal{X} \phi(X, y, \mathcal{X}) \supset \forall X \leq y \exists \mathcal{Z} Q_1\tilde{a_1}...Q_n\tilde{a_n} \psi(\tilde{a_1}, ..., \tilde{a_n}, \mathcal{Z}^{[1][\tilde{a_1}]...[\tilde{a_n}]}, X, y, \mathcal{Z}^{[2]}).$$

We may now appeal to the inductive hypothesis once more and apply the current theorem to the right-hand side of the previous implication, which results in

$$\forall X \leq y \exists \mathcal{X} \phi(X, y, \mathcal{X}) \supset \exists \mathcal{Z} \forall X \leq y Q_1 \tilde{a_1} ... Q_n \tilde{a_n} \psi(\tilde{a_1}, ..., \tilde{a_n}, \mathcal{Z}^{[X][1][\tilde{a_1}]...[\tilde{a_n}]}, X, y, \mathcal{Z}^{[X][2]}).$$

By applying $\Sigma_0^{\mathcal{B}}$-3COMP we can separate in two along the second “co-ordinate” the object $\mathcal{Z}$, quantified in the right-hand side:

$$\forall X \leq y \exists \mathcal{X} \phi(X, y, \mathcal{X}) \supset \exists \mathcal{X} \exists \mathcal{Z} \forall X \leq y Q_1 \tilde{a_1} ... Q_n \tilde{a_n} \psi(\tilde{a_1}, ..., \tilde{a_n}, \mathcal{Z}^{[X][\tilde{a_1}]...[\tilde{a_n}]}, X, y, \mathcal{X}^{[X]}).$$

The formula

$$\exists \mathcal{X} \forall X \leq y Q_1 \tilde{a_1} ... Q_n \tilde{a_n} \exists \mathcal{Z} \psi(\tilde{a_1}, ..., \tilde{a_n}, \mathcal{Z}, X, y, \mathcal{X}^{[X]})$$

is a logical consequence of the right-hand side of the previous implication and so we have proved

$$\forall X \leq y \exists \mathcal{X} \phi(X, y, \mathcal{X}) \supset \exists \mathcal{X} \forall X \leq y \phi(X, y, \mathcal{X}^{[X]}),$$

as required. □

The following is an immediate, useful corollary:

Corollary 6.3. *Let $\phi \in \Sigma_1^{\mathcal{B}}$. Then there exists $\psi \in$ strict$\Sigma_1^{\mathcal{B}}$ such that $W_1^1 \vdash \phi \leftrightarrow \psi$.*

Proof Sketch. Starting from the prenex normal form for ϕ, we may apply the theorem to obtain a provably equivalent formula with all the third-order quantifiers at the front, all of them existential. Then, as in line 6.2 in the proof of the theorem, by applying $\Sigma_0^{\mathcal{B}}$-3COMP these may be joined into one, obtaining an equivalent strict $\Sigma_1^{\mathcal{B}}$-formula, as desired. □

7 Definability in the Theories

First we define several versions of what it means for a function to be definable in a theory, in the context of this document:

Definition 7.1. *Let T be a theory with vocabulary $\mathcal{L} \supseteq \mathcal{L}_A^3$ and Φ a set of $\mathcal{L}$-formulas. Then a function $\tilde{f} \in \mathbb{E}^1 \cup \mathbb{E}^2$ is* **Φ-definable** *in T if there is some $\phi \in \Phi$ such that:*

1. *$T \vdash \forall \overline{x}, \overline{X}, \overline{\mathcal{X}} \exists! \tilde{y} \phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, \tilde{y})$*
2. *$\phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, \tilde{f}(\overline{x}, \overline{X}, \overline{\mathcal{X}}))$ is true in the standard model for all values of the parameters.*

The defining axiom for $\tilde{f}$ is then

$$\tilde{f}(\overline{x}, \overline{X}, \overline{\mathcal{X}}) = \tilde{y} \leftrightarrow \phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, \tilde{y})$$

Now, for superstring-valued functions we must use a slightly weaker kind of definability. This is because there is no way to bound a superstring, and thus no (bounded) way to assert the equality of two superstrings. Furthermore, our comprehension axioms assert the existence of certain superstrings but specify only an initial segment of their bits. Thus the following definitions:

Definition 7.2. *For superstring variables $\mathcal{X}$ and $\mathcal{Y}$ and term t, let $\mathcal{X} =_t \mathcal{Y}$ abbreviate the $\Sigma_0^{\mathcal{B}}$-formula $\forall Z \leq t(\mathcal{X}(Z) \leftrightarrow \mathcal{Y}(Z))$, where Z does not occur in t.*

Definition 7.3. *Let T, $\mathcal{L}$ and Φ be as above. Let $t(\overline{x}, \overline{X})$ be a number term over $\mathcal{L}$, which in the standard model bounds the* **lengths of bit-indices** *in the output of a function $\mathcal{F}(\overline{x}, \overline{X}, \overline{\mathcal{X}}) \in \mathbb{E}^3$. Then $\mathcal{F}$ is* **length 2^t Φ-definable in** *T if there is some $\phi \in \Phi$ such that:*

1. $T \vdash \forall \overline{x}, \overline{X}, \overline{\mathcal{X}} \exists \mathcal{Y} \phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, \mathcal{Y})$
2. $T \vdash \forall \overline{x}, \overline{X}, \overline{\mathcal{X}}, \mathcal{Y}, \mathcal{Y}' [\phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, \mathcal{Y}) \wedge \phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, \mathcal{Y}') \longrightarrow \mathcal{Y} =_t \mathcal{Y}']$
3. $T \vdash \forall \overline{x}, \overline{X}, \overline{\mathcal{X}}, \mathcal{Y}, \mathcal{Y}' [\mathcal{Y} =_t \mathcal{Y}' \wedge \phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, \mathcal{Y}) \longrightarrow \phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, \mathcal{Y}')]$
4. $\phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, \mathcal{F}(\overline{x}, \overline{X}, \overline{\mathcal{X}}))$ *is true in the standard model for all values of the parameters.*

The defining axiom for $\mathcal{F}$ is then

$$(\mathcal{Y} =_t \mathcal{F}(\overline{x}, \overline{X}, \overline{\mathcal{X}})) \leftrightarrow \phi(\overline{x}, \overline{X}, \overline{\mathcal{X}}, \mathcal{Y})$$

If $\mathcal{F}$ is length 2^t Φ-definable in T for some true bound $t(\overline{x}, \overline{X})$, then we say $\mathcal{F}$ is Φ-definable in T.

In sum, a length 2^t-definable function in a theory T is provably total, provably unique up to 2^t bits, and the formula defining the graph of the function is provably insensitive to bits beyond this bound. Furthermore, the true value of the function satisfies the graph. The defining axiom specifies the bits of the value of the function up to length 2^t, but beyond this point it is undefined (although also not relevant, as far as the graph is concerned).

Note that for suitable theories, a length 2^t-definable function $\mathcal{F}$ is also length 2^s-definable for any s that is provably larger than t, either by modifying the defining formula to check that the extra bits are zeroes, or by composing the function with an extender function that adds the extra zeroes. This will be discussed further in section 8.

Note further that a superstring-valued function definable in a theory in the sense of the previous definition can be conservatively added to the theory in the same way as the more concretely definable functions of the previous definition. The defining axiom is deliberately vague about specifying the value of the function, and therefore any (say) string-valued function G defined using (i.e., as a function of) $\mathcal{F}$ must provably depend only on the bits of the output of $\mathcal{F}$ actually fixed by the defining axiom – otherwise the uniqueness clause of the definition of G will presumably not hold.

It may be desirable in some cases to assert that a definable function with a superstring argument is insensitive in this way to variations in its superstring argument. The following definition formalizes this concept:

Definition 7.4. *Let $\tilde{f}(\mathcal{X}, \overline{y}, \overline{Y}, \overline{\mathcal{Y}}) \in \mathbb{E}^1 \cup \mathbb{E}^2$ be a definable function of a theory T (as above) and $t(\overline{y}, \overline{Y})$ a term of $\mathcal{L}$, the language of T. Then $\tilde{f}$ is* **insensitive to $\mathcal{X}$ beyond** *t if*

$$T \vdash \mathcal{X} =_t \mathcal{X}' \longrightarrow \tilde{f}(\mathcal{X}, \overline{y}, \overline{Y}, \overline{\mathcal{Y}}) = \tilde{f}(\mathcal{X}', \overline{y}, \overline{Y}, \overline{\mathcal{Y}})$$

Similarly, a function $\mathcal{F}(\mathcal{X}) \in \mathbb{E}^3$ that is length 2^s definable in T is **insensitive to $\mathcal{X}$ beyond** *t if*

$$T \vdash \mathcal{X} =_t \mathcal{X}' \longrightarrow \mathcal{F}(\mathcal{X}) =_s \mathcal{F}(\mathcal{X}')$$

If $\tilde{f}(\overline{y}, \overline{Y}, \mathcal{Y}_1, ..., \mathcal{Y}_k)$ has k superstring arguments and is $t_i(\overline{y}, \overline{Y})$-insensitive to $\mathcal{Y}_i$ for each I, then we say that $\tilde{f}$ is $(t_1, ..., t_k)$-insensitive.

We know that W^1_1 can $\Sigma^{\mathcal{B}}_0$-define all functions (of string variables) from the polynomial-time hierarchy. In fact, W^1_1 can $\Sigma^{\mathcal{B}}_1$-define all string functions computable in polynomial space:

Theorem 7.5. *Let $F \in FPSPACE^+ \cap \mathbb{E}^2$. Then there is a strict $\Sigma_1^{\mathcal{B}}$-formula ϕ such that F is $\Sigma_1^{\mathcal{B}}$-definable in W_1^1 by ϕ.*

XXX: Expand to deal with all argument, output types

Proof. Let F be any polynomial-space computable function of polynomial growth rate and let M be a PSPACE Turing machine computing F and $s(X)$ be a (number) term bounding the space used by M (including M's output tape) on input X (and thus also bounding the logarithm of the running time). Let $\phi_M(\mathcal{X}, Y, Z, i)$ state that $\mathcal{X}$ is a computation of M of length 2^i steps, with initial configuration coded by Y and final configuration coded by Z. $\mathcal{X}$ is stored as an array of configurations, indexed by configuration number expressed as a string, and for the sake of simplicity ϕ_M enforces that all configurations are the same size. $\phi_M(\mathcal{X}, Y, Z, i)$ states that for each configuration number smaller than 2^i (bounded second order universal quantifier) the corresponding configuration is valid and results from the previous one by one step of M (Σ_0^B subformula). Thus ϕ_M is $\Sigma_0^{\mathcal{B}}$. For concreteness in what follows, we shall reason in the sequent formulation $\mathrm{LK}^3 - W_1^1$ of W_1^1.

W_1^1 can clearly then prove

$$\forall X \leq |S| \exists \mathcal{X} \exists Y \leq |X| \phi_M(\mathcal{X}, X, Y, 0).$$

Now, W_1^1 proves

$$\phi_M(\mathcal{Y}, A, B, i), \phi_M(\mathcal{Z}, B, C, i) \longrightarrow \exists \mathcal{X} \exists Y \leq |A| \phi_M(\mathcal{X}, A, Y, i+1),$$

since $\Sigma_0^{\mathcal{B}}$-3COMP can be used to produce the third-order object $\mathcal{X}$ which consists of $\mathcal{Y}$ and $\mathcal{Z}$ spliced together, and W_1^1 can subsequently prove that such $\mathcal{X}$ satisfies ϕ_M as shown. Now being careful of the order in which we do so, we may introduce quantifiers in this sequent as follows, using the fact that all configurations in a computation are the same size: First

$$\phi_M(\mathcal{Y}, A, B, i), \exists \mathcal{X} \exists Y \leq |B| \phi_M(\mathcal{X}, B, Y, i) \longrightarrow \exists \mathcal{X} \exists Y \leq |A| \phi_M(\mathcal{X}, A, Y, i+1),$$

then adding a hypothesis to the succedent we obtain

$$\phi_M(\mathcal{Y}, A, B, i), \exists \mathcal{X} \exists Y \leq |B| \phi_M(\mathcal{X}, B, Y, i) \longrightarrow |A| \leq |S| \supset \exists \mathcal{X} \exists Y \leq |A| \phi_M(\mathcal{X}, A, Y, i+1),$$

then by reasoning about sizes of the configurations we may add a similar hypothesis to the second formula in the antecedent:

$$\phi_M(\mathcal{Y}, A, B, i), |B| \leq |S| \supset \exists \mathcal{X} \exists Y \leq |B| \phi_M(\mathcal{X}, B, Y, i) \longrightarrow \\ |A| \leq |S| \supset \exists \mathcal{X} \exists Y \leq |A| \phi_M(\mathcal{X}, A, Y, i+1),$$

and then introduce a quantifier like so:

$$\phi_M(\mathcal{Y}, A, B, i), \forall X \leq |S| \exists \mathcal{X} \exists Y \leq |X| \phi_M(\mathcal{X}, X, Y, i) \longrightarrow \\ |A| \leq |S| \supset \exists \mathcal{X} \exists Y \leq |A| \phi_M(\mathcal{X}, A, Y, i+1),$$

Then similar reasoning with the first formula in the antecedent yields

$$\forall X \leq |S| \exists \mathcal{X} \exists Y \leq |X| \phi_M(\mathcal{X}, X, Y, i), \forall X \leq |S| \exists \mathcal{X} \exists Y \leq |X| \phi_M(\mathcal{X}, X, Y, i) \longrightarrow \\ |A| \leq |S| \supset \exists \mathcal{X} \exists Y \leq |A| \phi_M(\mathcal{X}, A, Y, i+1).$$

Contraction and the introduction of a final quantifier in the succedent yields a sequent suitable for applying induction to:

$$\forall X \leq |S| \exists \mathcal{X} \exists Y \leq |X| \phi_M(\mathcal{X}, X, Y, i) \longrightarrow \forall X \leq |S| \exists \mathcal{X} \exists Y \leq |X| \phi_M(\mathcal{X}, X, Y, i+1),$$

$\forall^2 \Sigma_1^{\mathcal{B}}$-IND produces

$$\longrightarrow \forall X \leq |S| \exists \mathcal{X} \exists Y \leq |X| \phi_M(\mathcal{X}, X, Y, |X|).$$

Now it is easy to see that W_1^1 proves

$$\forall X \exists Y \exists \mathcal{X} \exists Z \leq s(|X|)|)(\phi_M(\mathcal{X}, \mathrm{I}nit_M(X), Z, s(|X|)) \wedge \mathrm{O}ut_M(Z, Y))$$

for a suitable function symbol $\mathrm{I}nit_M$ and formula $\mathrm{O}ut_M$, and $s(|X|)$ a term bounding the space used by M. Thus the existence part of the definability is obtained.

Point 2 of the definability is clear given the definition of ϕ_M from the Turing machine M computing f.

Finally, uniqueness is proved as follows: Firstly, define $\psi(k)$ to be the formula

$$\forall A \leq s(|X|) \forall B \leq s(|X|) \forall C \leq s(|X|) \forall Z \leq s(|X|)$$
$$(\phi_M(sub(\mathcal{X}, Z, k), A, B, k) \wedge \phi_M(sub(\mathcal{Y}, Z, k), A, C, k) \supset B = C),$$

where $sub(\mathcal{X}, Z, i)$ is a suitably defined functional which gives the subcomputation of $\mathcal{X}$ starting with configuration number encoded by Z and continuing for 2^i steps. $\psi(0)$ is provable in W_1^1 since the next configuration of a Turing machine is computable in linear time, and thus is definable even in V^1. $\psi(i) \supset \psi(i+1)$ is immediate, and so by ($\Sigma_0^{\mathcal{B}}$-)induction, $\psi(s(|X|))$, from which uniqueness follows.

□

Now we show that the same is true for $\widehat{W_1^1}$:

Theorem 7.6. *Let $F \in FPSPACE^+ \cap \mathbb{E}^2$. Then F is strict-$\Sigma_1^{\mathcal{B}}$-definable in $\widehat{W_1^1}$.*

XXX: Expand for types

Proof. Let ϕ_M be as in the previous theorem and let

$$\phi'_M(\mathcal{X}, i, S) \equiv \forall X \leq |S| \exists Y \leq |S| \phi_M(\mathcal{X}^{[X]}, X, Y, i).$$

This $\Sigma_0^{\mathcal{B}}$ formula expresses that $\mathcal{X}$ simultaneously encodes computations of M of length 2^i from every starting configuration of length $|S|$ to some ending configuration.

We now reason in $\mathrm{LK}^3 - \widehat{W_1^1}$. An $\mathcal{X}$ provably satisfying $\phi'_M(\mathcal{X}, 0, S)$ is computable in polynomial time (as a function in $\mathbb{E}^3$ of S) and so comprehension on a $\Sigma_0^{\mathcal{B}}$ formula suffices to obtain one. That it satisfies $\phi_M(\mathcal{X}, 0, S)$ follows easily so we have

$$\exists \mathcal{X} \phi'_M(\mathcal{X}, 0, S).$$

Now,

$$\exists \mathcal{X} \phi'_M(\mathcal{X}, i, S) \longrightarrow \exists \mathcal{Y} \phi'(\mathcal{Y}, i+1, S)$$

is proved by one application of $\Sigma_0^{\mathcal{B}}$-3COMP: given an $\mathcal{X}$ satisfying $\phi'_M(\mathcal{X}, i, S)$, a $\mathcal{Y}$ satisfying $\phi_M(\mathcal{Y}, i+1, S)$ is defined by a $\Sigma_0^{\mathcal{B}}(\mathcal{X})$-formula that for any requested initial configuration splices together the two relevant computations from the given $\mathcal{X}$.

The rest of the proof is analogous to that of the previous theorem. □

Now we show that all string functions from FEXP$^+$ are $\Sigma_1^{\mathcal{B}}$-definable in TTW_1^0. This is because TTW_1^0 proves $\Sigma_0^{\mathcal{B}}$-superstring-recursion:

$$\exists \mathcal{X} \phi^{\mathrm{rec}}(Y, \mathcal{X}),$$

where $\phi \in \Sigma_0^{\mathcal{B}}$, and

$$\phi^{\mathrm{rec}}(S, \mathcal{X}) \equiv \forall Y \leq |S|(\mathcal{X}(Y) \leftrightarrow \phi(Y, \mathcal{X}^{<Y})),$$

where $\mathcal{X}^{<Y}$ is a chop function.

Lemma 7.7. *TTW_1^0 proves the $\Sigma_0^{\mathcal{B}}$-superstring-recursion scheme*

Proof. Even TW_1^0 can prove (by induction on Y) that $\phi^{\mathrm{rec}}(Y, \mathcal{X}) \wedge \phi^{\mathrm{rec}}(Y, \mathcal{Z})$ implies the bits of $\mathcal{X}$ and $\mathcal{Z}$ are equal up to number Y. Now, following the analogous exposition from [4] for TV^0, define

$$\phi^{\mathrm{lessrec}}(S, \mathcal{X}) \equiv \phi^{\mathrm{rec}}(S, \mathcal{X}) \vee [\exists Y \leq |S|(L(Y, S) \wedge \phi^{\mathrm{rec}}(Y, \mathcal{X}) \wedge \neg \mathcal{X}(Y) \wedge \phi(Y, \mathcal{X}^{<Y}))],$$

stating that either $\phi^{\mathrm{rec}}(S, \mathcal{X})$, or that some prefix of $\mathcal{X}$ is correct up to position Y, where $\phi(Y, \mathcal{X}^{<Y})$, yet $\neg \mathcal{X}(Y)$. In other words, if $Rev(S, \mathcal{X})$ is a function symbol reversing the first S bits of $\mathcal{X}$, then $Rev(S, \mathcal{X})$ is "less than" the unique string $\mathcal{Y}$ satisfying $\phi^{\mathrm{lessrec}}(S, Rev(S, \mathcal{Y}))$.

Reasoning in TTW_1^0, $\phi^{\mathrm{lessrec}}(S, \mathcal{Z})$ holds, where $\mathcal{Z}$ is a null object, by induction on S. Now define $\mathcal{W}$ by $\forall X \leq |S|[\mathcal{W}(X) \leftrightarrow 1]$. $\mathcal{W}$ is an "all-ones" superstring, and if $\phi^{\mathrm{lessrec}}(S, \mathcal{W})$, then $\phi^{\mathrm{rec}}(S, \mathcal{W})$ (which would complete the proof), so assume $\neg \phi^{\mathrm{lessrec}}(S, \mathcal{W})$. By $\Sigma_0^{\mathcal{B}}$-SSIND for $\phi^{\mathrm{lessrec}}(S, Rev(S, \mathcal{X}))$, (since Rev fixes both $\mathcal{Z}$ and $\mathcal{W}$ above),

$$\exists \mathcal{X} \exists \mathcal{Y} \phi^{\mathrm{lessrec}}(S, Rev(S, \mathcal{X})) \wedge S_3(\mathcal{X}, \mathcal{Y}, S) \wedge \neg \phi^{\mathrm{lessrec}}(S, Rev(S, \mathcal{Y})).$$

Now TW_1^0 proves by induction on S that $\phi^{\mathrm{rec}}(S, Rev(S, \mathcal{X}))$. □

Now we show how to define EXP-time computations using $\Sigma_0^{\mathcal{B}}$-superstring-recursion:

Theorem 7.8. *Let $F \in FEXP^+ \cap \mathbb{E}^2$. Then F is strict-$\Sigma_1^{\mathcal{B}}$-definable in TTW_1^0.*

XXX: Expand for types

Proof Sketch. Let F be as described and M an exponential-time Turing machine computing F, with $s(X)$ a number term bounding the output size of M on input X and $t(X)$ bounding the logarithm of the run-time. We describe a formula $\phi_M(X, Y, \mathcal{X})$ suitable for applying the above recursion scheme to. The argument X denotes the input to M, and Y and $\mathcal{X}$ are as in the recursion scheme. $\phi_M(X, Y, \mathcal{X}) \equiv \phi_M^1(X, Y, \mathcal{X}) \vee \phi_M^2(X, Y, \mathcal{X})$.

The disjunct ϕ_M^1 evaluates to the appropriate bit of the initial configuration of M on input X if Y is small enough.

The other disjunct ϕ_M^2 is as follows: For every two positions W_1, W_2 smaller than Y, if $L_2(W_1, W_2)$ and W_1 and W_2 point to the start of configurations in $\mathcal{X}$, and there is no W_3 between them also pointing to the start of a configuration, then M computes bit Y to be 1. Bit Y is a function of a constant number of bits preceding Y and preceding $W_1 +_2 (Y -_2 W_2)$, where '$+_2$' and '$-_2$' are intended to represent arithmetic on strings.

ϕ_M as outlined is then clearly $\Sigma_0^{\mathcal{B}}$ (in fact, even $\Pi_2^{\mathcal{B}}$).

Now, applying the $\Sigma_0^{\mathcal{B}}$-superstring-recursion, $TTW_1^0 \vdash \exists \mathcal{X} \phi_M^{\mathrm{rec}}(X, Y, \mathcal{X})$, so for a suitable formula Out_M, $TTW_1^0 \vdash \forall X \exists Y (\exists \mathcal{X} (\phi_M^{\mathrm{rec}}(X, (2^{t(X)})^2, \mathcal{X}) \wedge Out_M(\mathcal{X}, (2^{t(X)})^2, Y)))$. Point 2 of the definability is clear, and uniqueness follows directly from the (bounded) uniqueness of superstrings satisfying ϕ_M^{rec}.

□

Now we show that HW_1^0 $\Sigma_1^{\mathcal{B}}$-defines the PSPACE functions:

Theorem 7.9. *Let $F \in FPSPACE^+ \cap \mathbb{E}^2$. Then F is strict-$\Sigma_1^{\mathcal{B}}$-definable in HW_1^0.*
XXX: Expand for types

Proof. Let F be as described and M a polynomial-space Turing machine computing F, with $s(X)$ a number term bounding the space used by M on input X. The idea now is to use a superstring $\mathcal{X}$ to encode a sequence of adjacency matrices. The ith matrix will indicate, for every pair of configurations of $F(X)$, if one yields the other in time at most 2^i. Furthermore, these matrices will alternate with unused space, so that the halfrecursion scheme can be applied.

Therefore we now describe a formula $\phi_M(X, Y, \mathcal{X})$ suitable for application of the halfrecursion scheme. X is the input to M and Y and $\mathcal{X}$ are as in the recursion scheme, and as before, $\phi_M(X, Y, \mathcal{X}) \equiv \phi_M^1(X, Y, \mathcal{X}) \vee \phi_M^2(X, Y, \mathcal{X})$.

The disjunct ϕ_M^1 is $|Y| = 2s(X) \wedge \exists Z, W(Y = Z \frown W \wedge (\mathrm{N}ext_M(Z, W) \wedge Z = W))$. This subformula ignores $\mathcal{X}$ and directly computes bit Y, if Y is the right length to indicate a pair of configurations.

The other disjunct ϕ_M^2 is as follows:

$$\exists A, B, C \leq |Y|(2|A| = 2|B| = 2|C| = |Y| - 2 \wedge Y = 0^2 \frown A \frown C \wedge \mathcal{X}(A \frown B) \wedge \mathcal{X}(B \frown C)).$$

This subformula verifies that Y is 00 followed by some pair of equal-length strings, and furthermore that 2 steps in the previous adjacency matrix in $\mathcal{X}$ yield the transition coded by Y.

So ϕ_M is $\Sigma_0^{\mathcal{B}}$ and $HW_1^0 \vdash \exists \mathcal{X} \phi_m^{\mathrm{hrc}}(X, Y, \mathcal{X})$. Let $\mathrm{O}ut_M$ be a suitable formula extracting the output from the least accepting configuration reachable from the starting configuration of $F(X)$, as coded in $\mathcal{X}$. Then $\mathrm{O}ut_M$ is clearly $\Sigma_0^{\mathcal{B}}$ and $HW_1^0 \vdash \forall X \exists Y(\exists \mathcal{X}(\phi_M^{\mathrm{hrc}}(X, 2^{6s(X)}, \mathcal{X}) \wedge \mathrm{Out}_M(\mathcal{X}, X, Y)))$, satisfying the existence part of the definability. Point 2 is clear and uniqueness is as in the previous theorem, using the provable (bounded) uniqueness of superstrings satisfying the recursion schemes. □

8 A Universal Conservative Extension of HW_1^0

In this section we define and develop $\overline{HW_1^0}$, a universal theory intended to be a conservative extension of HW_1^0. We loosely follow similar constructions of universal theories for P, NL and so on from [4] and [?]; however, our situation is considerably more complex as we have an additional sort (superstrings), and furthermore objects of that sort are unbounded.

We start with several additional function symbols beyond those provided in $\mathcal{L}_A^3$: recall the function f_{SE}, with open defining axioms SE' and SE", used as open replacements for the string extensionality axiom SE. B12' and B12" are open replacements for B12 defining the function *pd*. The superstring stretch function $\varsigma(a, b, \mathcal{X})$ is intended to return a superstring with the initial 2^{a+b} bits fixed such that the first 2^a of them agree with the input $\mathcal{X}$, and the remainder are zeroes. The open defining axiom is:

$$|Y| \leq a + b \supset (\varsigma(a, b, \mathcal{X})(Y) \leftrightarrow |Y| \leq a \wedge \mathcal{X}(Y)).$$

All these functions are $\Sigma_0^{\mathcal{B}}$-definable in W_1^0 (length s^{a+b}-definable in the case of ς).

The open theory $\overline{HW_1^0}$ we define below defines a succession of function symbols $\mathcal{L}_{PS}$ inductively from previous ones, and specifies a set of defining axions for each one. This requires more care than in analogous constructions because of the unbounded nature of our third-order objects, and the limited sense in which superstring-valued functions can be definable. For this reason, we will associate with each function and predicate symbol a polynomial (i.e., term in $\mathcal{L}_A^3$) which will be an upper-bound on its sensitivity to its third-order arguments, as a function of its other arguments. Each function symbol will be provably in HW_1^0 insensitive to its third-order arguments beyond this bound, in the sense of definition 7.4. To function symbols we will additionally associate a polynomial ($\mathcal{L}_A^3$-term) bounding the output as a function of the number and string arguments. Each number- or string-valued function symbol will be definable in HW_1^0 and provably bounded by its associated polynomial t, while superstring-valued function symbols will be length-2^t-definable. These terms will be explicitly written into the names of all function symbols defined below.

The following definition shows how to extend these sensitivity and bounding polynomials to certain terms and open formulas. It also identifies a class of open formulas called **permissible formulas** that are constructed with sufficient interleavings of the superstring stretch function to ensure their value is well defined, and hence suitable for use in defining new function symbols:

Definition 8.1. *a) The bounding polynomials for 0, 1, $x+y$, $x*y$ and $|X|$ are respectively 0, 1, $x+y$, $x*y$ and $|X|$. The sensitivity polynomials for these function symbols are all 0.*

b) The bounding polynomials of $pd(x)$ and $f_{SE}(X,Y)$ are x and $|X|$ respectively, and sensitivity polynomials for both are 0.

c) The superstring stretch function $\varsigma(a,b,\mathcal{X})$ has sensitivity a and bound $a+b$.

XXX: Not necessary? Only used as part of the construction.

d) If $\tilde{f}$ is a function symbol (of any type) with sensitivity s and bound t, $\mathcal{R}_1,...,\mathcal{R}_k$ are superstring terms with sensitivity $u_1,...,u_k$ and bound $v_1,...,v_k$, and finally $\tilde{h}_1,...,\tilde{h}_j$ are number or string terms with sensitivity $p_1,...,p_j$ and bounds $q_1,...,q_j$, then (assuming it is syntactically correct)

$$\tilde{f}(\varsigma(v_1,s(q_1,...,q_j),\mathcal{R}_1),...,\varsigma(v_k,s(q_1,...,q_j),\mathcal{R}_k),\tilde{h}_1,...,\tilde{h}_k)$$

has sensitivity $u_1+...+u_k+p_1+...+p_j+s(q_1,...,q_k)$ and bound $t(q_1,...,q_k)$.

e) $x \in Y$, $X \in \mathcal{Y}$, $x \leq y$, $x=y$ and $X=Y$ are permissible formulas with sensitivity polynomials 0, $|X|$, 0, 0 and 0, respectively.

f) If $\mathcal{R}$ is a term of sensitivity s and bound t, and H is a string term of sensitivity u and bound v, then $X \in \varsigma(t,|X|,\mathcal{R}))$ and $H \in \varsigma(t,v,\mathcal{R})$ are permissible formulas with sensitivities s and $s+u$ respectively.

g) Any other application of a predicate symbols to terms of the appropriate type results in a permissible formula whose sensitivity is the sum of the sensitivities of the given terms.

h) If ϕ and θ are permissible formulas of sensitivity s and t, then $\phi\wedge\theta$, $\phi\vee\theta$ and $\neg\phi$ are permissible with sensitivity $s+t$, $s+t$ and s, respectively.

Definition 8.2 ($\mathcal{L}_{PS}$). *$\mathcal{L}_{PS}$ is the smallest class satisfying*

a) $\mathcal{L}_{PS}$ includes $\mathcal{L}_A^3 \cup \{pd, <_2, f_{SE}, \varsigma\}$.

b) For each permissible open formula $\alpha(z,\overline{x},\overline{X},\overline{\mathcal{X}})$ of sensitivity $s(z)$ over $\mathcal{L}_{PS}$ and number term t over $\mathcal{L}_A^3$, there is a string function $F_{\alpha,t,s(t)}$ of sensitivity $s(t)$ and bound t with defining axiom

$$F_{\alpha,t,s(t)}(\overline{x},\overline{X},\overline{\mathcal{X}})(z) \leftrightarrow z<t \wedge \alpha(z,\overline{x},\overline{X},\overline{\mathcal{X}}) \tag{8.1}$$

intended to simulate 2-COMP.

c) *For each permissible open formula $\alpha(z, \overline{x}, \overline{X}, \overline{\mathcal{X}})$ of sensitivity $s(z)$ over $\mathcal{L}_{PS}$ and number term t over $\mathcal{L}^3_A$(free variables among those of α), there is a number function $g_{\alpha,t,s(t)}$ of sensitivity $s(t)$ and bound t with defining axioms*

$$g_{\alpha,t,s(t)}(\ldots) \leq t(\ldots) \tag{8.2}$$

$$g_{\alpha,t,s(t)}(\ldots) < t(\ldots) \supset \alpha(g_{\alpha,t,s(t)}(\ldots), \ldots) \tag{8.3}$$

$$z < g_{\alpha,t,s(t)}(\ldots) \supset \neg\alpha(z, \ldots) \tag{8.4}$$

intended to allow elimination of number quantifiers. It follows from these defining axioms that

$$\exists z < t\alpha(z, \ldots) \leftrightarrow g_{\alpha,t,s(t)}(\ldots) < t.$$

A suitable witness for these axioms is $g_{\alpha,t,s(t)}(\ldots) = \min z < t\alpha(z, \ldots)$.

d) *For each permissible open formula $\alpha(Z, \ldots)$ of sensitivity $s(|Z|)$ over $\mathcal{L}_{PS}$ and number term t over $\mathcal{L}^3_A$(free variables among those of α), there is a superstring function $\mathcal{F}_{\alpha,t,s(t)}$ of sensitivity $s(t)$ and bound t with defining axiom*

$$|Z| \leq t \supset [\mathcal{F}_{\alpha,t,s(t)}(\ldots)(Z) \leftrightarrow \alpha(Z, \ldots)] \tag{8.5}$$

intended to simulate 3-COMP.

e) *For each permissible open formula $\alpha(Z, \overline{x}, \overline{X}, \overline{\mathcal{X}})$ of sensitivity $s(|Z|)$ over $\mathcal{L}_{PS}$ and number term t over $\mathcal{L}^3_A$(free variables among those of α, there is a string function $G_{\alpha,t,s(t)}$ of sensitivity $s(t)$ and bound t with defining axioms*

$$|G_{\alpha,t,s(t)}(\ldots)| \leq t(\ldots) \tag{8.6}$$

$$|G_{\alpha,t,s(t)}(\ldots)| < t(\ldots) \supset \alpha(G_{\alpha,t,s(t)}(\ldots), \ldots) \tag{8.7}$$

$$Z <_2 G_{\alpha,t,s(t)}(\ldots) \supset \neg\alpha(Z, \ldots) \tag{8.8}$$

intended to allow elimination of string quantifiers. It follows that

$$\exists Z < t\alpha(Z, \ldots) \leftrightarrow |G_{\alpha,t,s(t)}(\ldots)| < t,$$

and a suitable witness is $G_{\alpha,t,s(t)}(\ldots) = \min Z < t\alpha(Z, \ldots)$

f) *For each three functions $\mathcal{G}(\ldots), \mathcal{H}(x, \mathcal{Z}, \ldots)$ and $l(x, \ldots)$ of $\mathcal{L}_{PS}$ with sensitivities $s_{\mathcal{G}}, s_{\mathcal{H}}$ and s_l and bounds $t_{\mathcal{G}}, t_{\mathcal{H}}$ and t_l there is a function $\mathcal{F}_{\mathcal{G},\mathcal{H},l}$ of sensitivity $s_{\mathcal{G}} + s_{\mathcal{H}} + s_l$ and bound t_l with defining axioms*

$$|Y| \leq l(0, \ldots) \supset [\mathcal{F}_{\mathcal{G},\mathcal{H},l}(0, \ldots)(Y) \leftrightarrow \varsigma(t_{\mathcal{G}}, t_l, \mathcal{G}(\ldots))(Y)] \tag{8.9}$$

$$|Y| \leq l(x+1, \ldots) \supset [\mathcal{F}_{\mathcal{G},\mathcal{H},l}(x+1, \ldots)(Y) \leftrightarrow \varsigma(t_{\mathcal{H}}, t_l, \mathcal{H}(x, \mathcal{F}_{\mathcal{G},\mathcal{H},l}(x, \ldots), \ldots))(Y)] \tag{8.10}$$

intended to define $\mathcal{F}_{\mathcal{G},\mathcal{H},l}$ by limited recursion from $\mathcal{G}$ and $\mathcal{H}$ with limit l.

Definition 8.3. *$\overline{HW^0_1}$ is the universal theory over $\mathcal{L}_{PS}$ consisting of the universal closures of B1-B11, B12' and B12" (open replacements for B12 defining pd), L1, L2, SE' and SE" (the defining axioms of f_{SE}), the defining axiom of ς, and finally all defining axioms 8.2–8.10 of $\mathcal{L}_{PS}$.*

After the lemma, we show that $\overline{HW^0_1}$ extends HW^0_1.

Lemma 8.4. *For every $\Sigma_0^{\mathcal{B}}$ formula ϕ there is an open formula α of $\mathcal{L}_{PS}$ such that $\overline{HW_1^0} \vdash \phi \leftrightarrow \alpha$.*

Proof outline. This follows by structural induction on ϕ, using cases c and e of the definition of $\mathcal{L}_{PS}$. The permissibility of the open formulas constructed is not a factor, as no superstring-valued functions are constructed. □

Theorem 8.5. $\overline{HW_1^0} \vdash HW_1^0$

Proof. B12 follows from B12' and B12". That $\Sigma_0^{\mathcal{B}}$-{2,3}COMP are provable follows from the previous lemma and cases b and d of the definition of $\mathcal{L}_{PS}$.

Finally, for any given $\phi \in \Sigma_0^{\mathcal{B}}$, $\mathcal{L}_{PS}$ contains a function witnessing the ϕ-ss-hrc scheme: this function is defined by limited recursion from a function that outputs $\mathcal{X}$ satisfying

$$\forall Y \leq z + 1(\mathcal{X}(Y) \leftrightarrow \phi(Y, \mathcal{X}^{Y/2}))$$

given $\mathcal{X}'$ satisfying

$$\forall Y \leq z(\mathcal{X}'(Y) \leftrightarrow \phi(Y, \mathcal{X}'^{Y/2})).$$

The correctness of this function is proved by open($\mathcal{L}_{PS}$)-IND, which is derived in the standard way in $\overline{HW_1^0}$ from the comprehension. XXX: Expand □

To show that $\overline{HW_1^0}$ is conservative over HW_1^0, we inductively show that every function of $\mathcal{L}_{PS}$ is $\Sigma_1^{\mathcal{B}}$-definable in HW_1^0. In fact, this seems not to be a strong enough induction hypothesis, so we in fact show something stronger: that each function symbol of $\mathcal{L}_{PS}$ is $\Sigma_0^{\mathcal{B}}$-HR-definable, a concept that we now define:

Definition 8.6. *Let T be a theory over $\mathcal{L} \supseteq \mathcal{L}_A^3$ and Φ a set of $\mathcal{L}$-formulas. Then a function $\tilde{f}(\overline{x}, \overline{X}, \overline{\mathcal{X}}) \in \mathbb{E}^1 \cup \mathbb{E}^2$ is* **Φ-HR-definable in** *T if there are $\phi_1(Y, \mathcal{Z}, \overline{x}, \overline{X}, \overline{\mathcal{X}}), \phi_2(\tilde{y}, \mathcal{Z}, \overline{x}, \overline{X}, \overline{\mathcal{X}}) \in \Phi$ and term $s(\overline{x}, \overline{X})$ over $\mathcal{L}$ (all free variables displayed) such that*

1. *$T \vdash \forall \overline{x}, \overline{X}, \overline{\mathcal{X}} \exists! \tilde{y} \exists \mathcal{Z}(\phi_1^{hrc}(s(...), \mathcal{Z}, \overline{x}, \overline{X}, \overline{\mathcal{X}}) \wedge \phi_2(\tilde{y}, \mathcal{Z}^{<s(\dots)}, \overline{x}, \overline{X}, \overline{\mathcal{X}}))$*
2. *$\tilde{f}(\overline{x}, \overline{X}, \overline{\mathcal{X}})$ satisfies the defining formula in the standard model for all values of the parameters.*

Similarly, $\mathcal{F}(\overline{x}, \overline{X}, \overline{\mathcal{X}}) \in \mathbb{E}^3$ is **length $2^{t(\overline{x}, \overline{X})}$ Φ-HR-definable in** *T if there are $\phi_1, \phi_2 \in \Phi$ and s over $\mathcal{L}$ (as above) such that*

1. *$T \vdash \forall \overline{x}, \overline{X}, \overline{\mathcal{X}} \exists \mathcal{Y} \exists \mathcal{Z}(\phi_1^{hrc}(s(...), \mathcal{Z}, \overline{x}, \overline{X}, \overline{\mathcal{X}}) \wedge \forall Y \leq t(\overline{x}, \overline{X})(\mathcal{Y}(Y) \leftrightarrow \phi_2(Y, \mathcal{Z}^{<s(\dots)}, \overline{x}, \overline{X}, \overline{\mathcal{X}})))$*
2. *$\mathcal{F}(\overline{x}, \overline{X}, \overline{\mathcal{X}})$ satisfies the defining formula in the standard model for all values of the parameters.*

If $\mathcal{F}$ is length 2^t Φ-HR-definable in T for some true bound t, then we say $\mathcal{F}$ is Φ-HR-definable in T.

Some explanation of the previous definition is in order. A function $\tilde{f}$ that is Φ-HR-definable in a theory T is defined syntactically by a Φ-halfrecursion (computing a superstring) composed with a Φ-definition. In the case that $\Phi = \Sigma_0^{\mathcal{B}}$ and $T = HW_1^0$, this corresponds to defining a superstring with a halfrecursion operation from a PH predicate, and composing with a PH function to produce the final value. These two operations composed in this way can produce every function in FPSPACE$^+$, as the halfrecursion operation can produce the computation of a PSPACE Turing Machine (or an array of computations if a superstring-valued function is to be computed), and then a PH function can extract the output of the machine (or collect the bits of the superstring value of the function from the array of computations).

Now the conservativity of $\overline{HW_1^0}$ over HW_1^0 follows from the following lemma:

Lemma 8.7. *The functions of $\mathcal{L}_{PS}$ are all $\Sigma_0^{\mathcal{B}}$-HR-definable in HW_1^0. Furthermore, they are all provably insensitive to their superstring arguments past the claimed sensitivity bounds.*

Proof. This is proved by induction on the definition of the functions, considered in some appropriate enumeration. At each step, the $\Sigma_0^{\mathcal{B}}$-HR-definitions of all relevant function symbols are combined into one $\Sigma_0^{\mathcal{B}}$-HR-definition of the new function symbol.

Consider for example a function symbol $\mathcal{F}$ defined from a permissible formula α and $\mathcal{L}_A^3$-term t using case d) of the definition of $\mathcal{L}_{PS}$. By the induction hypothesis, each function symbol in α is $\Sigma_0^{\mathcal{B}}$-HR-definable in HW_1^0. Now all the formulas ϕ_1 and ϕ_2 from the $\Sigma_1^{\mathcal{B}}$-HR-definitions of these functions symbols can be combined into one $\Sigma_0^{\mathcal{B}}$ formula ϕ, such that ϕ^{hrc} asserts the existence of one large superstring computing the value of $\mathcal{F}$. This large superstring contains subcomputations for each occurrence of a function symbol in α, arranged in some suitable order of evaluation; each such subcomputation is followed by another phase extracting the output of the corresponding function symbol occurrence from the computation. Bounds on the lengths of these computations and outputs are all known in advance, so the ϕ_1 and ϕ_2 formulas can be amended to reference their respective parts of this big superstring. Finally, a $\Sigma_1^{\mathcal{B}}$-formula extracts the result from the end of this large computation. Since α is permissible, at every step the computation is provably well defined (i.e., provably depends only on bits of superstring outputs actually fixed by defining axioms of the appropriate function symbols). Thus by induction on the structure of α, $\mathcal{F}$ is uniquely defined to the given bound.

Case b) is similar. Cases c) and e) use the function symbols from cases b) and d) respectively to construct a table of values of α on all inputs up to the given bound and then extract the minimum value satisfying α. The defining axioms are then proved in HW_1^0 directly from the definition of this table.

Finally, a function symbol $\mathcal{F}$ defined using case f) is computed by limited recursion. Again, one large superstring records, one after the other, the computations of each step of this recursion (and there are polynomially many). The $\Sigma_0^{\mathcal{B}}$-HR-definition of $\mathcal{F}$ asserts that this superstring exists and is defined appropriately, and the value of $\mathcal{F}$ is extracted by a $\Sigma_0^{\mathcal{B}}$-bit-definition. □

Theorem 8.8 (KPT Witnessing for HW_1^0). *Let $\phi(X, \mathcal{Y}, Z)$ be a $\Sigma_0^{\mathcal{B}}$-formula such that $HW_1^0 \vdash \forall X \exists \mathcal{Y} \forall Z \phi(X, \mathcal{Y}, Z)$. Then there are functions $\mathcal{F}_1, ..., \mathcal{F}_k \in FPSPACE^+$ such that*

$$HW_1^0 \vdash \forall X \forall \overline{Z}[\phi(X, \mathcal{F}_1(X), Z_1) \vee ... \vee \phi(X, \mathcal{F}_k(X, Z_1, ..., Z_{k-1}), Z_k)]$$

Proof Sketch. The standard model-theoretic argument applies using $\overline{HW_1^0}$. □

9 Sequent Calculus Formulations

In this section we introduce some equivalent sequent formulations of several theories. LK^3 is like the system LK, but with the addition of the following quantifier introduction rules:

$$\forall : \textbf{left} \quad \frac{\phi(\tilde{Y}), \Gamma \longrightarrow \Delta}{\forall \tilde{X} \phi(\tilde{X}), \Gamma \longrightarrow \Delta} \qquad \text{and} \qquad \exists : \textbf{right} \quad \frac{\Gamma \longrightarrow \Delta, \phi(\tilde{Y})}{\Gamma \longrightarrow \Delta, \exists \tilde{X} \phi(\tilde{X})}$$

and

$$\exists : \textbf{left} \quad \frac{\phi(\tilde{Y}), \Gamma \longrightarrow \Delta}{\exists \tilde{X} \phi(\tilde{X}), \Gamma \longrightarrow \Delta} \qquad \text{and} \qquad \forall : \textbf{right} \quad \frac{\Gamma \longrightarrow \Delta, \phi(\tilde{Y})}{\Gamma \longrightarrow \Delta, \forall \tilde{X} \phi(\tilde{X})}$$

where $\tilde{X}$ and $\tilde{Y}$ are either both second- or both third-order variables, and in the latter two rules $\tilde{Y}$ may not occur in the conclusion of the inference. Formally, LK^3 also adopts the usual conventions concerning free and bound variables, as in [2].

The system $\mathrm{LK}^3 - W_1^i$ additionally includes the $\forall^2\Sigma_i^{\mathcal{B}}$-IND rule:

$$\frac{\Gamma, \phi(b) \longrightarrow \phi(b+1), \Delta}{\Gamma, \phi(0) \longrightarrow \phi(t), \Delta},$$

where b appears only as indicated and $\phi \in \forall^2\Sigma_i^{\mathcal{B}}$. As initial sequents we allow all substitution instances of the axioms (other than induction) of W_1^1. Note that all rules of $\mathrm{LK}^3 - W_1^i$ are valid in W_1^i, and furthermore, $\mathrm{LK}^3 - W_1^i$ proves the induction and comprehension schemes of W_1^i.

The system $\mathrm{LK}^3 - \widehat{W_1^i}$ is as above, but with the $\Sigma_i^{\mathcal{B}}$-IND rule instead.

The system $\mathrm{LK}^3 - HW_1^0$ is LK^3 plus, as initial sequents, all substitution instances of axioms of HW_1^0.

10 Witnessing Theorems

10.1 A Witnessing Theorem for W_1^1

In this section we prove a Buss-style witnessing theorem showing that every $\Sigma_1^{\mathcal{B}}$-definable string function of W_1^1 is computable in PSPACE.

The standard definition of an anchored cut in LK^3 is extended in the usual way for $\mathrm{LK}^3 - W_1^1$ by allowing cuts on the descendents of principal formulas of the $\forall^2\Sigma_1^{\mathcal{B}}$-IND rule, in addition to cuts on descendents of formulas in nonlogical axioms. The anchored completeness theorem for LK^3 can be extended to $\mathrm{LK}^3 - W_1^1$ in the usual way to cope with the induction rules, as detailed in [8].

With this in mind, we can now state the witnessing theorem we wish to prove:

Theorem 10.1. *Suppose $W_1^1 \vdash \exists Y\phi(X,Y)$, for $\phi(X,Y) \in \Sigma_1^{\mathcal{B}}$ with all free variables displayed. Then there exists a function $f \in$ PSPACE of polynomial growth rate such that for every string X, $\phi(X, f(X))$ is true.*

Before we prove the theorem, we shall need several definitions:

Definition 10.2. *Let $\psi \equiv \forall X \leq t\exists\mathcal{X}\phi(X,\mathcal{X}) \in \forall^2\Sigma_1^{\mathcal{B}}$, with other free variables not shown. Consider an assignment to the free variables of ψ. Then the string relation $\mathcal{A}(A,B)$* **satisfies** *ψ (with respect to the assignment to the free variables of ψ) iff for every string A of no more than t bits, $\phi(A, \{B\}(\mathcal{A}(A,B)))$ is true in the standard model, where $\{B\}(\mathcal{A}(A,B))$ denotes the predicate on strings obtained by fixing to A the first argument to the relation $\mathcal{A}$.*

Definition 10.3. *Let S be the sequent $\Gamma \longrightarrow \Delta$ such that $\Gamma \bigcup \Delta \subset \forall^2\Sigma_1^{\mathcal{B}}$, i.e.*

$$\Gamma = \{\forall A_i \leq s_i \exists\mathcal{A}_i\gamma_i(A_i, \mathcal{A}_i, \overline{\mathcal{B}}, \overline{B}, \overline{b})\}$$

and

$$\Delta = \{\forall C_i \leq t_i \exists\mathcal{C}_i\delta_i(C_i, \mathcal{C}_i, \overline{\mathcal{B}}, \overline{B}, \overline{b})\},$$

with $\{\gamma_i\}\bigcup\{\delta_i\} \subset \Sigma_0^{\mathcal{B}}$, and although we write for simplicity the initial string and third-order quantifiers for each formula, in fact for some of the formulas either the initial string quantifier or both initial quantifiers may be absent.

Then **PSPACE Oracle Witnessing Operators (POWOs)** *for S are operators, or type-2 predicates. For each formula from Δ*

$$\forall C_i \leq t_i \exists \mathcal{C}_i \delta_i(C_i, \mathcal{C}_i, \overline{\mathcal{B}}, \overline{B}, \overline{b})$$

which is not $\Sigma_0^{\mathcal{B}}$ (and may or may not have the leading string quantifier as pictured), the POWO f_i is a predicate with arguments $\{\overline{\mathcal{B}}, \overline{B}, \overline{b}\}$ (for the free variables of the sequent), $\{\mathcal{A}_j(A_j, X)\}$ (for the string relations satisfying the formulas in the antecedent) and finally $\{C_i, X\}$, making f_i into a two-place string relation when the other arguments are fixed. The f_i must have the property that for any assignment to the free variables $\overline{\mathcal{B}}, \overline{B}, \overline{b}$ of S and string relations $\{\mathcal{A}_j(A_j, X)\}$, if each formula γ_j is satisfied by the corresponding $\mathcal{A}_j$, then some δ_i is satisfied by the string relation $\{C_i, X\}f_i$, obtained by fixing all but the last two arguments to the operator f_i.

Furthermore, each f_i is computable by an oracle Turing machine in space (including on the query tapes) polynomial in the lengths of its string and number inputs.

Now the theorem will follow from the following lemma:

Lemma 10.4. *Suppose $LK^3 - W_1^1 \vdash \Gamma \longrightarrow \Delta$, where $\Gamma \bigcup \Delta \subset \forall^2\Sigma_1^{\mathcal{B}}$. Then there exist PSPACE Oracle Witnessing Operators for $\Gamma \longrightarrow \Delta$.*

Proof of Theorem 10.1 from Lemma 10.4. Suppose $W_1^1 \vdash \exists Y \phi(X, Y)$, for $\phi(X, Y) \in \Sigma_1^{\mathcal{B}}$ with all free variables displayed. By Parikh's theorem, $W_1^1 \vdash \exists Y \leq t(|X|)\phi(X, Y)$, for some term t. By Corollary 6.3, $W_1^1 \vdash \phi(X, Y) \leftrightarrow \exists \mathcal{Y} \psi(X, Y, \mathcal{Y})$, for some $\psi \in \Sigma_0^{\mathcal{B}}$. Also, $W_1^1 \vdash \exists Y \leq t(|X|)\exists \mathcal{Y}\psi(X, Y, \mathcal{Y}) \leftrightarrow \exists \mathcal{Y} \exists Y \leq t(|X|_q)\psi(X, Y, \mathcal{Y})$. Applying the lemma to the sequent $\longrightarrow \exists \mathcal{Y} \exists Y \leq t(|X|)\psi(X, Y, \mathcal{Y})$, we obtain a PSPACE (in $|X|$) predicate for $\mathcal{Y}$ satisfying that sequent, and so for particular X the string Y can be obtained in PSPACE by evaluating ψ, with access to the predicate $\mathcal{Y}$, on each string of length $\leq t(|X|)$ in turn. It is easy to see that the computed string Y satisfies $\phi(X, Y)$ (for the same fixed X). □

All that remains is to prove the lemma:

Proof of Lemma 10.4. Suppose $LK^3 - W_1^1 \vdash \Gamma \longrightarrow \Delta$, where $\Gamma \bigcup \Delta \subset \forall^2\Sigma_1^{\mathcal{B}}$, and consider an anchored proof π of this sequent. Since both the endsequent of π and every nonlogical axiom of $LK^3 - W_1^1$ is $\forall^2\Sigma_1^{\mathcal{B}}$, and since the induction rule is limited to this same class of formulas, every formula in π is $\forall^2\Sigma_1^{\mathcal{B}}$.

We now show by induction on the number of sequents in π that POWOs exist for $\Gamma \longrightarrow \Delta$.

Base Case: The base case is that $\Gamma \longrightarrow \Delta$ is either an initial sequent of LK^3 or an instance of an axiom. The only such sequents requiring POWOs are those with a third-order quantifier in the succedent, namely an instance

$$\longrightarrow (\exists \mathcal{Y})(\forall Z \leq s(\overline{B}, \overline{b}))[\phi(\overline{\mathcal{B}}, \overline{B}, \overline{b}, Z) \leftrightarrow \mathcal{Y}(Z)]$$

of $\Sigma_0^{\mathcal{B}}$-3COMP, where $\phi \in \Sigma_0^{\mathcal{B}}$, subject to the restriction that $\mathcal{Y}$ does not occur free in ϕ. The only POWO required for this sequent is computed by the predicate

$$f(\overline{\mathcal{B}}, \overline{B}, \overline{b}, \overline{\mathcal{A}}, Z) \leftrightarrow |Z| \leq s(\overline{B}, \overline{b}) \wedge \phi(\overline{\mathcal{B}}, \overline{B}, \overline{b}, Z),$$

which is in some level of the polynomial-time hierarchy, and thus certainly in PSPACE.

Induction Step: The induction step has several cases depending on which rule has been used to derive $\Gamma \longrightarrow \Delta$.

1. Weakening:

   The POWOs from the hypothesis are modified to take any extra arguments the new formula introduces (free variables or an existential third-order quantifier in the antecedent) and to ignore them. If the formula is added to the succedent and contains a third-order quantifier, a constant-false predicate taking the appropriate arguments is added as the new POWO for the conclusion.

2. Contraction:

   If the contraction occurs in the succedent on a formula ϕ with a third-order quantifier, then one less POWO is required for the conclusion. Construct a new POWO for ϕ which evaluates ϕ on each original POWO in turn (each evaluation is computable in PSPACE) and then behaves like whichever satisfies ϕ, if any. This computation requires only a constant number of bits more than the maximum of the space used by the two original POWOs.

   If the contraction occurs in the antecedent on a formula ϕ with a third-order quantifier, then all original POWOs must be modified to accept one less oracle argument. Each is modified to query the original POWO but now passing the oracle argument from ϕ twice.

3. Exchange, introduction of $\neg$, $\vee$ on the right and $\wedge$ on the left:

   These rules can neither introduce nor eliminate free variables. No third-order quantifiers are added or removed, and no formula with a third-order quantifier is changed, so the POWOs from the hypothesis are used without modification for the conclusion.

4. Introduction of $\vee$ on the left and $\wedge$ on the right:

   These inferences have two hypotheses, and the principal formula is $\Sigma_0^{\mathcal{B}}$ and so needs no POWO. Any side formula which is not $\Sigma_0^{\mathcal{B}}$ will have a POWO for each hypothesis. As in the case of contraction, the POWO for such a formula in the conclusion evaluates the formula on each POWO from the hypotheses, and then simulates whichever satisfies it, if any.

5. First- or second-order $\forall :$ **left** and $\exists :$ **right**:

   The conclusion of such an inference may have less free variables than the hypothesis. Taking for example an $\exists :$ **right** inference with principal formula $\exists X \phi(X)$ with the corresponding formula in the hypothesis being $\phi(B)$ and B not free in the conclusion, all POWOs for the hypothesis will have B as an argument. If this argument is fixed to the empty string, the resulting set of POWOs will suffice for the conclusion of the inference (unless $\phi \notin \Sigma_0^{\mathcal{B}}$, addressed below). $\forall :$ **left** is similar and in the first-order cases one analogously substitutes 0 for eliminated variables.

   If $\phi \notin \Sigma_0^{\mathcal{B}}$ then the principal formula of the inference is $\forall A_i \leq s_i \exists \mathcal{A}_i \gamma_i(A_i, \mathcal{A}_i, \overline{\mathcal{B}}, \overline{B}, \overline{b})$ and occurs in the antecedent. In addition to the procedure above (substituting the empty string for the eliminated free string variable), the POWOs must be modified so that any query $\mathcal{A}_i(X)$ becomes $\mathcal{A}_i(\lambda, X)$, adding the empty string as an additional argument, since in the conclusion this oracle argument to the POWOs is two-place.

6. First- or second-order $\forall :$ **right** and $\exists :$ **left**:

   As in the previous case free variables are eliminated by such inferences. However, it is not sufficient to substitute a dummy value for them as above since such a value would not witness the new quantifier properly. For example, if the new quantifier is universal on the right and the principal formula is false under some assignment, the POWOs (from the hypothesis) for the remaining formulas expect a value falsifying the principal formula. This value is found by exhaustive search, evaluating the formula on each possible value of the new quantifier (subject to the bound). The POWOs for the conclusion perform this search before querying the POWOs from the hypothesis. The extra search is clearly carried out in polynomial space.

If the principal formula is not $\Sigma_0^{\mathcal{B}}$, then it is $\forall C_i \leq t_i \exists \mathcal{C}_i \delta_i(C_i, \mathcal{C}_i, \overline{\mathcal{B}}, \overline{B}, \bar{b})$ and is in the succedent. In this one special case the POWO for δ_i retains the same number of arguments in the conclusion, due to the string quantifier preceding the third-order quantifier. The POWO for δ_i alone is not modified as above, but instead passes the new argument, C_i, to the POWO from the hypothesis, in place of the eliminated free variable.

7. Third-order $\exists$: **left**:

   The principal formula is $\exists \mathcal{A}_i \gamma_i(A_i, \mathcal{A}_i, \overline{\mathcal{B}}, \overline{B}, \bar{b})$. All POWOs from the antecedent are modified to accept oracle argument $\mathcal{A}_i$ instead of the free third-order variable eliminated by the quantifier introduction.

8. Third-order $\exists$: **right**:

   If the eigenvariable $\mathcal{B}$ occurs in the lower sequent, then the POWO for the principal formula is defined by
   $$f(\overline{\mathcal{B}}, \overline{B}, \bar{b}, \overline{\mathcal{A}}, Z) \leftrightarrow \mathcal{B}(Z)$$

   If not, analogously to the lower-order cases of this rule, the new quantifier is witnessed by any value and thus the POWO for the new quantifier may ignore its arguments and always return false. Furthermore, a constant-false predicate is supplied in the place of the eliminated variable as an argument to the other POWOs from the hypothesis.

9. The **cut** rule:

   The inference is
   $$\frac{\Gamma \longrightarrow \phi, \Delta \qquad \Gamma, \phi \longrightarrow \Delta}{\Gamma \longrightarrow \Delta}.$$

   A POWO for the conclusion proceeds in two phases: First, it evaluates its formula using the POWO from the left hypothesis, and if that POWO satisfies the formula, it emulates it. Otherwise, it emulates the POWO from the right hypothesis, and uses the POWO for ϕ from the left hypothesis to supply a value for the oracle argument. The whole procedure uses at most the sum of the space requirements of the two POWOs from the hypotheses.

   If any free variables are eliminated, then as before a dummy argument of the correct type is supplied to the POWOs.

10. $\forall^2\Sigma_1^{\mathcal{B}}$-IND:

    The inference is:
    $$\frac{\Gamma, \phi(b) \longrightarrow \phi(b+1), \Delta}{\Gamma, \phi(0) \longrightarrow \phi(t), \Delta}.$$

    The POWOs for the conclusion will iterate the construction from the previous case, as the current instance of the induction rule could be simulated by t instances of the cut rule, along with some weakenings.

More precisely, let f_ϕ be the POWO for the instance of ϕ in the succedent of the hypothesis. Let ψ be any formula in the succedent of the hypothesis (including ϕ) and f_ψ its POWO. We construct a POWO f'_ψ for ψ in the conclusion in stages:

$f^0_\psi(X, Y) \leftrightarrow f_\psi(X, Y)$.

$f^k_\psi(X, Y) \leftrightarrow (\psi(f^{k-1}_\psi) \wedge f^{k-1}_\psi(X, Y)) \vee (\neg\psi(f^{k-1}_\psi) \wedge f^{k-1}_\psi(f_\phi, X, Y))$.

f^1_ψ checks if f_ψ satisfies ψ and if so, simulates f_ψ. If not, f^1_ψ computes $f_\psi(f_\phi)$, that is to say, uses f_ϕ to answer queries to the oracle argument corresponding to ϕ.

f^k_ψ checks if f^{k-1}_ψ satisfies ψ and if so, simulates f^{k-1}_ψ. If not, f^k_ψ computes $f^{k-1}_\psi(f_\phi)$.

f'_ψ, then, evaluates t and computes f^t_ψ. Computing f^t_ψ requires t times the space required to compute f_ϕ plus the space requirements of f_ψ, and so only increases the space usage of POWOs by a polynomial factor.

□

10.2 Witnessing for HW^0_1

Theorem 10.5. *Suppose $HW^0_1 \vdash \exists Y\phi(X, Y)$, for $\phi(X, Y) \in \Sigma^{\mathcal{B}}_1$ with all free variables displayed. Then there exists a function $F \in FPSPACE^+$ for every string X, $\phi(X, f(X))$ is true.*

Proof. This theorem is proved analogously to the previous theorem. The witnessing lemma required is in fact simpler, as all formulas in the anchored proof will be $\Sigma^{\mathcal{B}}_1$. All cases of the previous witnessing lemma are the same for the present one, except of course the induction rule is now much more restricted. We need one additional case for the $\Sigma^{\mathcal{B}}_0$-superstring-halfrecursion scheme. A witnessing operator for the superstring quantifier $\exists\mathcal{X}$ on an instance of this scheme computes a requested bit of $\mathcal{X}$ by evaluating the $\Sigma^{\mathcal{B}}_0$ formula ϕ from the scheme, and recursively computing the bits of $\mathcal{X}$ required by ϕ. Modulo the recursive calls, this computation is clearly in the PH. Now, the depth of the recursion is only polynomial, as each recursive call halves the relevant number of bits of $\mathcal{X}$. This entire recursive procedure is thus computable in polynomial space. □

11 Translation into BPLK

In this section we define a translation of $\Sigma^{\mathcal{B}}_0$ formulas in the language $\mathcal{L}^3_A$of W^1_1 (i.e. possibly with free third-order variables, but no third-order quantifiers) into families of propositional sequents in the language of Boolean programs. We prove a lemma implying that if $\phi(A) \in \Sigma^{\mathcal{B}}_\infty$ and if $W^1_1 \vdash \phi(A)$ then BPLK has short proofs of the translations of ϕ. BPLK is from [6].

First, we can extend the definitions of a Boolean Program and of a BPLK proof as follows:

Definition 11.1. *A* **Boolean semiprogram** *is like a Boolean program, except we allow that some function symbols used in the program be undefined ("free").*

Definition 11.2. *A* **BPLK-sequence** *is the same as a BPLK proof except that the requirement that all function symbols occurring in the sequence be defined by the accompanying Boolean program is dropped. Furthermore, the accompanying Boolean program is instead a Boolean semiprogram. Any undefined function symbol appearing in the sequence or the semiprogram is called "free".*

The following translation is defined for the larger class $\Sigma^{\mathcal{B}}_0$ and is necessary for the main lemma in the proof:

Definition 11.3. *Let $\phi(\mathcal{A}_1, ..., \mathcal{A}_j, A_1, ..., A_k)$ be $\Sigma_0^{\mathcal{B}}$ in the language $\mathcal{L}_A^3$. For every $m_1, ..., m_k$ we construct a Boolean semiprogram $P_\phi^{m_1,...,m_k}$ and a formula $||\phi||^{m_1,...,m_k}$ in the language of Boolean programs, with the atoms $\overline{\overline{p}} = (\overline{p_i}, i = 1, ..., k)$, where each $\overline{p_i} = (p_{i,0}, ..., p_{i,m_k})$. By induction on the structure of ϕ:*

- *If ϕ is the atomic formula $s = t$ then s and t are first-order terms with no free first-order variables. Third-order variables do not appear in first-order terms so all variable occurrences in s and t are of the form $|A_i|$ for some second-order variable A_i. Then using the value m_i for this subterm the terms s and t can be evaluated to $\underline{s}$ and $\underline{t}$. We define $||s = t||^{m_1,...,m_k} := 1$ if $\underline{s} = \underline{t}$ and $||s = t||^{m_1,...,m_k} := 0$ otherwise. The semiprogram $P_\phi^{m_1,...,m_k} := \emptyset$.*
- *The case for $\phi \equiv t \leq s$ is similar.*
- *If ϕ is the atomic formula $t \in_2 A_i$ then we can as above evaluate t and then $||\phi||^{m_i} := p_{i,\underline{t}}$ if $\underline{t} \leq m_i$ and $||\phi||^{m_i} := 0$ otherwise. $P_\phi^{m_i} := \emptyset$.*
- *If ϕ is the atomic formula $A_i \in \mathcal{A}_j$ then $||\phi||^{m_i} := g_{\mathcal{A}_j}(p_{i,0}, ..., p_{i,m_i})$. $P_\phi^{m_i} := \emptyset$. The intention is that $g_{\mathcal{A}_j}$ be a free function symbol and we shall be careful not to add a definition for any function symbol of this form to our Boolean semiprograms. Furthermore, this is the only case in the construction where a free function symbol is produced.*
- *If $\phi \equiv \neg\psi$ then $||\phi||^{m_1,...,m_k} := \neg||\psi||^{m_1,...,m_k}$ and $P_\phi^{m_1,...,m_k} := P_\psi^{m_1,...,m_k}$.*
- *If $\phi \equiv \psi \circ \xi \quad (\circ \in \{\wedge, \vee\})$, then $||\phi||_{m_1,...,m_k} := ||\psi||^{m'_1,...,m'_{k'}} \circ ||\xi||^{m''_1,...,m''_{k''}}$ and $P_\phi^{m_1,...,m_k} := P_\psi^{m'_1,...,m'_{k'}} \diamond P_\xi^{m''_1,...,m''_{k''}}$. Here the lists $\overline{m'}$ and $\overline{m''}$ are the sublists of $\overline{m}$ corresponding to which of the free variables of ϕ occur free in ψ and ξ, and the "$\diamond$" operator is the merging of Boolean semiprograms, defined in [5].*
- *If ϕ is $\exists x \leq t\psi(x)$ then $||\phi||^{m_1,...,m_k} := \bigvee_{n \leq \underline{t}} ||\psi(n)||^{m_1,...,m_k}$ ($\phi(n)$ is $\phi(x)[s/x]$ where s is a constant term of value n, say $\overbrace{1 + ... + 1}^{n}$). $P_\phi^{m_1,...,m_k} := P_\psi^{m_1,...,m_k}$.*
- *If ϕ is $\forall x \leq t\psi(x)$ then $||\phi||^{m_1,...,m_k} := \bigwedge_{n \leq \underline{t}} ||\psi(n)||^{m_1,...,m_k}$. $P_\phi^{m_1,...,m_k} := P_\psi^{m_1,...,m_k}$.*
- *If ϕ is $\exists X \leq t\psi(X)$ then $||\phi||^{m_1,...,m_k} := f_\phi(\overline{\overline{p}})$ and $P_\phi^{m_1,...,m_k}$ is as follows:*

$$f_{\phi,0}^l(\overline{\overline{p}}, q_0, ..., q_l) := ||\psi||^{m_1,...,m_k,l} \qquad \textit{for each } l \leq \underline{t}$$

$$f_{\phi,i}^l(\overline{\overline{p}}, q_i, ..., q_l) := f_{\phi,i-1}^l(\overline{\overline{p}}, 0, q_i, ..., q_l) \vee f_{\phi,i-1}^l(\overline{\overline{p}}, 1, q_i\cdot,,,.q_l) \qquad \textit{for each } l \leq \underline{t} \textit{ and } i \leq l+1$$

$$f_\phi(\overline{\overline{p}}) := \bigvee_{l \leq \underline{t}} f_{\phi,l+1}^l(\overline{\overline{p}})$$

- *If ϕ is $\forall X \leq t\psi(X)$ then $||\phi||^{m_1,...,m_k} := f_\phi(\overline{\overline{p}})$ and $P_\phi^{m_1,...,m_k}$ is as follows:*

$$f_{\phi,0}^l(\overline{\overline{p}}, q_0, ..., q_l) := ||\psi||^{m_1,...,m_k,l} \qquad \textit{for each } l \leq \underline{t}$$

$$f_{\phi,i}^l(\overline{\overline{p}}, q_i, ..., q_l) := f_{\phi,i-1}^l(\overline{\overline{p}}, 0, q_i, ..., q_l) \wedge f_{\phi,i-1}^l(\overline{\overline{p}}, 1, q_i\cdot,,,.q_l) \qquad \textit{for each } l \leq \underline{t} \textit{ and } i \leq l+1$$

$$f_\phi(\overline{\overline{p}}) := \bigwedge_{l \leq \underline{t}} f_{\phi,l+1}^l(\overline{\overline{p}})$$

It is clear that for fixed ϕ, the size of $||\phi||^{m_1,...,m_k}$ is polynomial in $m_1.,,,.m_k$. Whenever we talk of BPLK proofs or BPLK-sequences involving translations of this form, we shall insist that the associated Boolean (semi-)program extend the (semi-)program resulting from the translation.

The following lemma is the main lemma of the proof. In the previous section, since it is not possible to translate a general $\Sigma_1^{\mathcal{B}}$ formula into the language of BPLK, we defined POWOs and used them to witness a sequent containing third-order quantifiers. Similarly, in the lemma below we shall translate sequents with third-order quantifiers as if those third-order variables were free, and then show that BPLK can prove the existence of a function symbol witnessing the sequent in much the same way. For this to work it would ordinarily be necessary for the formulas all to be **strict** $\Sigma_1^{\mathcal{B}}$. Unfortunately that cannot be guaranteed since the induction scheme in W_1^1 is for slightly more general formulas. We shall address this problem by first rewriting sequents into the equivalent form given by the replacement theorem and then translating them into the language of Boolean programs.

Lemma 11.4. *Let $LK^3 - W_1^1 \vdash \Gamma \longrightarrow \Delta$ where $\Gamma \bigcup \Delta \subset \forall^2\Sigma_1^{\mathcal{B}}$, i.e.*

$$\Gamma = \{\forall A_i \leq s_i \exists \mathcal{A}_i \gamma_i(A_i, \mathcal{A}_i, \overline{\mathcal{B}}, \overline{B}, \overline{b})\}$$

and

$$\Delta = \{\forall C_i \leq t_i \exists \mathcal{C}_i \delta_i(C_i, \mathcal{C}_i, \overline{\mathcal{B}}, \overline{B}, \overline{b})\},$$

with $\{\gamma_i\} \bigcup \{\delta_i\} \subset \Sigma_0^{\mathcal{B}}$, and although we write for simplicity the initial string and third-order quantifiers for each formula, in fact for some of the formulas either the initial string quantifier or both initial quantifiers may be absent.

Then for each $m_1, ..., m_k$ and $n_1, ..., n_l$ there are function symbols $h_i^{\overline{m},\overline{n}}$ and BPLK-sequences with endsequents

$$\begin{aligned} &..., ||\forall A_i \gamma_i(A_i, \mathcal{A}_i^{[A_i]}, \overline{\mathcal{B}}, \overline{B}, \underline{\overline{n}})||^{m_1,...,m_k}, ... \\ &\quad \longrightarrow ..., ||\forall C_i \delta_i(C_i, \mathcal{C}_i^{[C_i]}, \overline{\mathcal{B}}, \overline{B}, \underline{\overline{n}})||^{m_1,...,m_k}[h_i^{\overline{m},\overline{n}}/g_{\mathcal{C}_i}], ... \end{aligned}$$

where $h_i^{\overline{m},\overline{n}}$ are called witnessing function symbols and are not free, but may be defined in terms of free function symbols (in particular, $g_{\mathcal{A}_i}$). Furthermore, these sequences have size polynomial in $m_1, ..., m_k$ and $n_1, ..., n_l$.

The notation $...[h_i^{\overline{m},\overline{n}}/g_{\mathcal{C}_i}]$ in the succedent means that one should first perform the translation, and then substitute function symbol h_i for the free symbol $g_{\mathcal{C}_i}$ in the result.

Proof. We show the existence of the desired BPLK-sequence by induction on the number of sequents in the W_1^1 proof, in a manner very similar to the witnessing theorem of the previous section. The witnessing function symbols of the present lemma are analogous to POWOs.

Base Case: This is trivial for initial sequents and the witnessing function symbol, if required, is defined to be the constant false predicate. For translations of axioms B1-B12, L1, L2 and instances of $\Sigma_0^{\mathcal{B}}$-2COMP, it follows from the analogous result for V_1^1 and Extended Frege. For translations of instances of $\Sigma_0^{\mathcal{B}}$-3COMP, the witnessing function symbol has defining formula identical to the comprehension formula, and then the translation of the instance is proved using the introduction rule for this symbol followed by repeated substitutions and $\wedge$: **right** inferences.

Induction Step: There are cases depending on the final inference of the W_1^1 proof:

1. Weakening, Exchange, introduction of $\neg$, $\vee$ on the right and $\wedge$ on the left:

   These cases are all either structural rules or not applicable to formulas with third-order quantifiers and thus the same rule is applied in the BPLK proof. In the case of weakening, the conclusion may have more free variables than hypothesis. In that case new witnessing function symbols are defined to ignore the new arguments and compute the same value as the old ones, and these must be substituted for the old ones (by induction on the structure of the formula it can easily be seen that BPLK can prove each formula equivalent to one with the new function symbols instead).

2. Contraction:, introduction of $\vee$ on the left and $\wedge$ on the right:

   The only obstacle to using the identical propositional rule is that the principal formula of a contraction inference and the side formulas of the two-hypothesis inferences have two ancestors which will in general be witnessed by different witnessing function symbols (if they occur in the succedent). The solution is to define new witnessing function symbols by cases and then for each affected formula prove that the translation witnessed by the new function symbol implies the disjunction of the translations witnessed by the two old symbols.

   For example, a side formula $\forall C_i \leq t_i \exists \mathcal{C}_i \delta_i(C_i, \mathcal{C}_i)$ with witnessing function symbols h'_i and h''_i would have new witnessing function symbol

   $$h_i := (||\delta_i(C_i, \mathcal{C}_i^{[C_i]})||[h'_i/g_{\mathcal{C}_i}] \wedge h'_i) \vee (||\delta_i(C_i, \mathcal{C}_i^{[C_i]})||[h''_i/g_{\mathcal{C}_i}] \wedge h''_i)$$

   in the conclusion.

3. Introduction of a first-order quantifier:

   These cases are handled by the introduction of the appropriate propositional connective (disjunction or conjunction). In the case of a universal quantifier on the right or of an existential one on the left, proofs for each value of the free variable are concatenated together. In the other cases the proof for the hypothesis is first extended by weakening to add the other disjuncts (conjuncts on the left).

4. Introduction of a second-order quantifier:

   These cases are handled the same way as in the simulation of G by BPLK, in that essentially a big disjunction or conjunction is constructed over all values of a set of propositional variables.

   Additionally, if the principal formula is $\forall C_i \leq t_i \exists \mathcal{C}_i \delta_i(C_i, \mathcal{C}_i)$, then more work is needed. First, a new witnessing function symbol is defined as follows:

   $$h'_i(\overline{p}, \overline{q}) := (\overline{p} = \overline{r} \wedge h_i(\overline{q}))$$

   where $\overline{r}$ are the propositional variables associated with C_i, $\overline{p}$ are precisely as numerous as $\overline{r}$ and $\overline{q}$ are the same variables as the arguments to the original h_i. Then, a derivation is inserted proving

   $$||\delta_i(C_i, \mathcal{C}_i)||[h_i/g_{\mathcal{C}_i}] \longrightarrow ||\delta_i(C_i, \mathcal{C}_i^{[C_i]})||[h'_i/g_{\mathcal{C}_i}].$$

   The second-order quantifier introduction is then handled as usual.

5. Introduction of a third-order quantifier:

   These cases are easy: On the left, this amounts to renaming the arguments to the witnessing function symbols (from free variables to a free function symbol) and on the right it means producing a new witnessing function symbol defined equivalent to the existing free function symbol for that variable and substituting it into the sequent.

6. Cut, Induction:

   The cut rule is handled by defining new witnessing function symbols for the conclusion by cases, using the witnessing function symbol for the cut formula. For induction this procedure is iterated as many times as the value of the induction bound.

   For example, if the cut formula is $\forall C_i \leq t_i \exists \mathcal{C}_i \delta_i(C_i, \mathcal{C}_i)$, then a new witnessing function symbol h_j for $\forall C_j \leq t_j \exists \mathcal{C}_j \delta_j(C_j, \mathcal{C}_j)$ would be defined as follows, where h'_j is the witnessing function symbol for the hypothesis with the cut formula on the right, and h''_j that for the hypothesis with the cut formula on the left:

$$h_j := (||\delta_j(C_j, \mathcal{C}_j^{[C_j]})||[h'_j/g_{\mathcal{C}_j}] \wedge h_j) \vee h''_j(h_i).$$

□

References

[1] S. Buss. *Bounded Arithmetic.* Bibliopolis, Naples, 1986.

[2] Samuel R. Buss, editor. *Handbook of Proof Theory.* Elsevier Science B. V., Amsterdam, 1998.

[3] S. A. Cook. CSC 2429S: Proof Complexity and Bounded Arithmetic. Course notes, URL: "http://www.cs.toronto.edu/∼sacook/csc2429h", Winter 2002.

[4] Stephen A. Cook. Theories for complexity classes and their propositional translations. Manuscript, 2004.

[5] Alan Skelley. Relating the PSPACE reasoning power of Boolean programs and quantified Boolean formulas. Master's thesis, University of Toronto, 2000. Available from ECCC in the 'theses' section.

[6] Alan Skelley. Propositional PSPACE reasoning with Boolean programs versus quantified Boolean formulas. In *ICALP*, volume 3142 of *Lecture Notes in Computer Science*, pages 1163–1175. Springer, 2004.

[7] Alan Skelley. A third-order bounded arithmetic theory for PSPACE. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, *CSL*, volume 3210 of *Lecture Notes in Computer Science*, pages 340–354. Springer, 2004.

[8] Michael Soltys. A model-theoretic proof of the completeness of LK proofs. Manuscript, available on author's web page, 1999.