Truth Definitions in Finite Models

Leszek Aleksander Kołodziejczyk
Institute of Philosophy
Warsaw University
Krakowskie Przedmieście 3, 00-047 Warsaw, Poland
e-mail: l.kolodziejczyk@zodiac.mimuw.edu.pl

October 18th, 2003

Abstract

The paper discusses the notion of *finite model truth definitions* (or *FM–truth definitions*), introduced by M. Mostowski as a finite model analogue of Tarski's classical notion of truth definition.

We compare FM–truth definitions with Vardi's concept of the *combined complexity* of logics, noting an important difference: the difficulty of defining FM–truth for a logic $\mathcal{L}$ does not depend on the syntax of $\mathcal{L}$, as long as it is decidable. It follows that for a natural $\mathcal{L}$ there exist FM–truth definitions whose evaluation is much easier than the combined complexiy of $\mathcal{L}$ would suggest.

We apply the general theory to give a complexity–theoretical characterization of the logics for which the Σ^d_m classes (prenex classes of higher order logics) define FM–truth. For any $d \geq 2, m \geq 1$ we construct a family $\{[\Sigma^d_m]^{\leq k}\}_{k \in \omega}$ of syntactically defined fragments of Σ^d_m which satisfy this characterization. We also use the $[\Sigma^d_m]^{\leq k}$ classes to give a refinement of known results on the complexity classes captured by Σ^d_m.

We close with a few simple corollaries, one of which gives a sufficient condition for the existence, given a vocabulary σ, of a fixed number k such that model checking for all first order sentences over σ can be done in deterministic time n^k.

1 Introduction

The basic task of finite model theory, showing that some languages are more expressive in the finite than others, has turned out to be extremely difficult.

One of the principal difficulties consists in the scarcity of available methods. In particular, it seems that many methods of classical model theory are useless in finite models.

Tarski's method of truth definitions ([T33]) ranks among the most classical methods of comparing the semantical strength of logics. The essence of this method lies in Tarski's famous theorem on the undefinability of truth. The theorem states, roughly speaking, that a logic $\mathcal{L}$ closed under some basic first order constructions, most notably negation, does not define arithmetical truth for itself, i.e. there is no $\mathcal{L}$–formula with one free variable which is true of (a Gödel number of) an $\mathcal{L}$–sentence exactly when the sentence itself is true (we are speaking of truth in the standard model of arithmetic, and are of course assuming that some appropriate arithmetization of syntax has been carried out). Now, if one proves that a logic $\mathcal{L}'$, which is known to be at least as expressive as $\mathcal{L}$, additionally defines truth for $\mathcal{L}$, then one has also shown — using the "method of truth definitions" — that $\mathcal{L}'$ is strictly more expressive that $\mathcal{L}$.

An attempt to adapt Tarski's method to the needs of finite model theory was made by M. Mostowski in [MM01] and [MM0?]. Mostowski introduced the notion of a *finite model truth definition* (*FM–truth definition*, for short) and proved a finite version of Tarski's theorem. He also obtained some results on the existence of FM–truth definitions for sublogics of finite order logic.

It seems difficult to estimate the potential significance of FM–truth definitions as a tool for separating logics in finite models. Due to the well known limitations of straightforward diagonal arguments in finite model theory (or complexity theory), it is very unlikely that any of the hard open problems on whether two given logics are equivalent in finite models can be solved by showing that one of the logics has an FM–truth definition for the other. But perhaps more likely is a situation in which it would be shown that one of the logics defines FM–truth for some third logic, and the other does not.

In this paper, we take a closer look at some aspects of the theory of FM–truth definitions. In particular, we try to compare the notion of FM-truth definition with an ealier, apparently related, concept introduced by Vardi in [V82]. Vardi defined the *combined complexity* of a logic $\mathcal{L}$ as the computational complexity of checking whether $\mathbf{M} \models \varphi$ for varying finite models $\mathbf{M}$ and $\mathcal{L}$–sentences φ — in other words, as the complexity of a "universal algorithm" for $\mathcal{L}$. We point out a key difference between FM–truth definitions and combined complexity, and later use this difference to characterize the situations in which one logic defines FM–truth for another.

The structure of the paper is as follows.

Section 2 is preliminary: we discuss the basic terminology and notation, explain the main ideas of the theory of FM–truth definitions, and introduce finite order logic $\mathcal{L}^\omega$, with which we work later on in section 5.

In section 3, we partially explain the relation between FM–truth definitions and combined complexity: if a (reasonable) logic $\mathcal{L}'$ captures the complexity class corresponding to the combined complexity of another logic $\mathcal{L}$, then $\mathcal{L}'$ defines FM–truth for $\mathcal{L}$. We also show a basic difference between the two notions: the combined complexity of a logic $\mathcal{L}$ is well known to depend essentially both on the semantics and the syntax of $\mathcal{L}$. However, the difficulty of defining FM–truth for $\mathcal{L}$ depends, perhaps surprisingly, only on the expressive power of $\mathcal{L}$, not on its syntax.

In section 4 we apply this last result to show that defining FM–truth for a logic is much easier than capturing the complexity class corresponding to the combined complexity of that logic. For example, the combined complexity of existential second order logic, Σ^1_1, is known to be nondeterministic exponential time, but there exist FM–truth definitons for Σ^1_1 which can be evaluated in nondeterministic time bounded by a "barely superpolynomial" function. We also explain the reason for this difference: it lies in the "asymptotic" nature of FM–truth definitions, i.e. in the fact that an FM–truth definition for a logic is allowed to misrepresent the logical value of any sentence of that logic in finitely many models, whereas a universal algorithm is always required to provide a correct answer.

Section 5 presents a case study: we look at the prenex classes of higher order logics (Σ^d_m), and characterize the logics for which they define FM–truth. A complexity–theoretical characterization of these logics follows trivially from the results of section 4, but we also discuss a concrete example of some syntactically defined subclasses of the Σ^d_m classes. In order to find FM–truth definitions for these subclasses, we refine known results on the complexity classes captured by prenex fragments of higher order logics (theorem 5.6).

In the final section 6, we formulate some simple corollaries of our results: among these, a sufficient condition for the existence — for any given vocabulary σ — of a fixed number k such that model checking for first order logic over σ can be performed in deterministic time n^k (or nondeterministic time n^k).

2 Preliminaries

2.1 Basic terminology and notation

We are going to work over finite models ("model" means "finite model") with built–in arithmetic. That is, we are going to assume, firstly, that the universes of our models are initial segments $n = \{0, \ldots, n-1\}$ of the natural numbers, secondly, that the arithmetical predicates $+, \times, BIT, \leq$ and constants $0, MAX$ are always in the vocabulary and are always interpreted in the standard way[1]. Thus, for example, MAX is to be intepreted as the largest element of the universe, and $+(a, b, c)$ is to hold if and only if $a + b = c$. $BIT(a, b)$ holds if and only if the b–th bit in the binary expansion of a (counting from the least significant bit) is 1.

All vocabularies are finite and relational, possibly with individual constants. We will refer to models of vocabulary σ as σ–models. Thus, for example, models of the empty (i.e. purely arithmetical) vocabulary $\emptyset$ are $\emptyset$–models. Obviously, for any n there is only one $\emptyset$–model of cardinality n. The universe of a model $\mathbf{M}$ is denoted by M. Observe that M is a number, and $card(M) = M$.

We use the term "logic" in a deliberately vague way (in particular, a logic does not have to be a logic in Lindström's sense, as we do not require e.g. closure under negation), but it is supposed to encompass all the logics commonly used in finite model theory. We do require that the syntax and the truth relation for any logic $\mathcal{L}$ be decidable (by the latter, we mean that the relation "$\mathbf{M} \models \varphi$" for varying models $\mathbf{M}$ and $\mathcal{L}$–sentences φ is to be decidable over every vocabulary).

If $\mathcal{L}$ and $\mathcal{L}'$ are logics, then $\mathcal{L} \leq \mathcal{L}'$ means that $\mathcal{L}'$ is at least as expressive as $\mathcal{L}$, i.e. all classes of models definable by an $\mathcal{L}$–sentence are also definable by an $\mathcal{L}'$–sentence. $\mathcal{L} \equiv \mathcal{L}'$ holds if $\mathcal{L} \leq \mathcal{L}'$ and $\mathcal{L}' \leq \mathcal{L}$. $\mathcal{L} < \mathcal{L}'$ holds if $\mathcal{L} \leq \mathcal{L}'$ but not $\mathcal{L} \equiv \mathcal{L}'$. For any vocabulary σ, $\mathcal{L} \leq_\sigma \mathcal{L}'$ means that $\mathcal{L}'$ is is at least as expressive as $\mathcal{L}$ over models of vocabulary σ. $\mathcal{L} \equiv_\sigma \mathcal{L}'$ and $\mathcal{L} <_\sigma \mathcal{L}'$ are defined accordingly.

Let K and K' be classes of σ–models. We say that K' is a finite variant of K if the symmetric difference of K and K' is finite.

We assume familiarity with standard computational complexity classes such as P, NP or PH. Recall the definitions of the two basic nondeterministic exponential time classes: $NETIME$ is $\bigcup_{c \in \omega} NTIME(2^{cn})$, $NEXPTIME$ is $\bigcup_{k \in \omega} NTIME(2^{n^k})$. It should be noted that we take the n in these definitions to stand for the cardinality of a model and not for the length of its description, which is usually polynomially larger (this is important in the case of classes such as $NETIME$). Another convention we freely use when discussing complexity classes is that $NTIME(f)$ stands for

[1] Many of these arithmetical predicates are in fact redundant (actually, BIT suffices to define all the others). Hence, including them is merely a question of convenience. An in–depth analysis of the definability of arithmetical relations in finite models may be found in [Z0?].

$NTIME(O(f))$ (and the same for other types of resources).

Some other, perhaps less known complexity classes will be defined in section 2.4.

If $\mathcal{L}$ is a logic and $\mathcal{C}$ is a complexity class, we say that $\mathcal{L}$ captures $\mathcal{C}$ if for any σ and any class of σ–models K, it holds that K is recognized by a Turing machine of the class $\mathcal{C}$ if and only if K is defined by an $\mathcal{L}$–sentence (i.e. there is an $\mathcal{L}$–sentence φ such that K consists of exactly those σ–models which satisfy φ). If any $\mathcal{L}$–definable class K is in $\mathcal{C}$, we say that model checking for $\mathcal{L}$ is in $\mathcal{C}$. If any class in $\mathcal{C}$ is $\mathcal{L}$–definable, $\mathcal{L}$ is said to capture at least $\mathcal{C}$.

2.2 Truth definitions in finite models

We describe, with some modifications, the "FM–truth definitions" framework built in [MM01] and [MM0?].

Assume that we have chosen a suitable finite alphabet A such that we may treat syntactical objects such as variables, formulae etc. (of any reasonable logic) as words over A. We fix some correspondence between words $w \in A^*$ and natural numbers $\ulcorner w \urcorner$ (their Gödel numbers). Via this correspondence, we are often going to treat the elements of our models (i.e. natural numbers) as elements of A^*.

To be definite, we may assume that the cardinality of A is a power of 2, e.g. 16, and consider each $a \in A$ as a binary string, of length 4 in our case. Then for a given $w \in A^*$, $w = w_1 \dots w_k$ with $w_i \in A$, we let $\ulcorner w \urcorner$ be the number whose binary representation has length $4k+1$ and is of the form $1w_1 \dots w_k$. In this way, many numbers do not correspond to words, but that is not a problem.

We want to represent syntactical relations between words in A^* (such as "x is a first order formula" or "x is the result of preceding the formula y by an existentially quantified monadic second order variable") in our models. Of course, no reasonable "representation" of an infinite relation within a single finite model is possible. Thus, we aim for representation in the sense of the following definition.

Definition 2.1 *A relation $R \subseteq \omega^n$ is FM–represented by the $\emptyset$–formula $\varphi(x_1, \dots, x_n)$ if any only if for any $a_1, \dots, a_n \in \omega$, $\varphi(a_1, \dots, a_n)$ is true in almost all $\emptyset$–models exactly if $R(a_1, \dots, a_n)$ holds, and false in almost all $\emptyset$–models if $R(a_1, \dots, a_n)$ does not hold. R is FM–representable if there is a formula which FM–represents it.*

The question now is: are syntactic relations, such as those mentioned above, FM–representable? Fortunately, for any reasonable, recursive language this is the case, because of the *FM–representability theorem*:

Theorem 2.2 ([MM01]) *A relation $R \subseteq \omega^n$ is FM–representable if and only if it is of degree $\leq \mathbf{0}'$ (recursive with an RE oracle). Every FM–representable relation is represented by a first order formula.*

Once we can represent syntax in finite models, we may start heading towards a finite model version of Tarski's theorem on the undefinability of truth. But first we must decide what a truth definition is to be. We agree on the following "asymptotic" notion:

Definition 2.3 *Let $\mathcal{L}$ be a logical language and σ be a vocabulary. We say that the σ–formula $Tr_{\mathcal{L},\sigma}(x)$ is an FM–truth definition for $\mathcal{L}$ over σ if and only if for every $\mathcal{L}$–sentence ψ of vocabulary σ,*

$$\mathbf{M} \models \psi \equiv Tr_{\mathcal{L},\sigma}(\ulcorner \psi \urcorner)$$

holds for almost all σ–models $\mathbf{M}$.

Thus, if we have an FM–truth definition for $\mathcal{L}$, then for a given $\mathcal{L}$–sentence ψ, there is a number n such that in all models of cardinality greater than n the definition "knows" whether ψ is true or not.

Of course, the classical version of Tarski's theorem is based on the famous diagonal lemma, which states that for any formula with one free variable there is a sentence which asserts of itself the property expressed by the formula. It turns out that there is a finite model version of this lemma:

Theorem 2.4 ([MM01]; Diagonal lemma, finite version) *For any σ, any logic $\mathcal{L}$ closed under first order quantification and under forming conjunctions with first order formulae, and any $\mathcal{L}$–formula $\varphi(x)$ of vocabulary σ with only one free (first order) variable x, there exists an $\mathcal{L}$–sentence ψ such that*

$$\mathbf{M} \models \psi \equiv \varphi(\ulcorner \psi \urcorner)$$

holds in almost all σ–models.

Finally, as an easy consequence of the diagonal lemma, we obtain a finite model version of Tarski's theorem:

Theorem 2.5 ([MM01]; Tarski's theorem, finite version) *If σ is a vocabulary and $\mathcal{L}$ is a logic closed under first order quantification, forming conjunctions with first order formulae, and negation, then there is no $\mathcal{L}$–formula which is an FM–truth definition for $\mathcal{L}$ over σ.*

When $\mathcal{L}, \mathcal{L}'$ are logics and there is an $\mathcal{L}'$–formula which is an FM–truth definition for $\mathcal{L}$ over σ, we say that $\mathcal{L}'$ FM–defines truth for $\mathcal{L}$ over σ and write $\mathcal{L} \ll_\sigma \mathcal{L}'$.

2.3 Combined complexity

In [V82], Vardi introduced three important notions of complexity associated with a logic $\mathcal{L}$. The *data complexity* of $\mathcal{L}$ is essentially the complexity of model checking for $\mathcal{L}$, i.e. the complexity of verifying whether $\mathbf{M} \models \varphi$ holds for varying models $\mathbf{M}$ and fixed $\mathcal{L}$–sentences φ. The *expression complexity* of $\mathcal{L}$ is, conversely, the complexity of checking $\mathbf{M} \models \varphi$ for varying φ and fixed $\mathbf{M}$. Finally, the *combined complexity* of $\mathcal{L}$ is the complexity of checking $\mathbf{M} \models \varphi$ in the case where both $\mathbf{M}$ and φ vary.

The data complexity of $\mathcal{L}$ depends exclusively on the expressive power of $\mathcal{L}$. On the other hand, the expression (and combined) complexity depends in general also on the syntax of $\mathcal{L}$. Vardi observed, however, that many natural logics obey a simple pattern: the expression and combined complexity are both one exponential level higher than the data complexity.

For example, the data complexity of existential second order logic, Σ^1_1, is in NP, while both the expression and the combined complexity of Σ^1_1 are NEXPTIME–complete. More precisely: a) Σ^1_1 captures NP; b) for any model $\mathbf{M}$ of cardinality at least 2, the problem of checking whether $\mathbf{M} \models \varphi$ for varying Σ^1_1 sentences φ is NETIME–complete and therefore NEXPTIME–complete; c) for any vocabulary σ, the problem of checking $\mathbf{M} \models \varphi$ for varying σ–models $\mathbf{M}$ and Σ^1_1 sentences φ is NEXPTIME–complete.

An in–depth analysis of sufficient conditions for a logic to obey this pattern, numerous examples of logics which do, and an example of a logic which does not, may be found in [GLV98].

In this paper, we will be particularily interested in the combined complexity of logics, as one might expect that FM–truth definitions for a logic should be somehow related to its combined complexity. After all, both notions are clearly linked, albeit in a slightly different way, to the idea of a "universal algorithm" for the logic.

To examine whether such a relationship indeed exists, we will need to treat the objects combined complexity deals with, i.e. ordered pairs consisting of a model and a formula, as single models — perhaps of a different vocabulary. There are of course many ways in which this can be done. We briefly describe the method we are going to use. Recall that the formulae of the logics we consider are oficially words over some fixed alphabet A (consisting, for example, of the 16 characters $0000, 0001, \dots, 1111$). For any vocabulary σ, let σ^+ be the expansion of σ by a unary predicate U and a number of additional unary predicates $P_{0000}, \dots, P_{1111}$ to represent the elements of A. An ordered pair $(\mathbf{M}, w)$, where $\mathbf{M}$ is a σ–model and $w \in A^*$, may now be thought of as the single σ^+–model $(\mathbf{M} + w)$ defined as follows. The universe of $(\mathbf{M} + w)$ is $M + lh(w)$. $U^{(\mathbf{M}+w)} = M$; thus, U divides $(\mathbf{M} + w)$ into an $\mathbf{M}$–part

$\{0, \ldots, M-1\}$ and a w–part $\{M, \ldots, M + lh(w) - 1\}$. The interpretations of the non–arithmetical σ–symbols agree with $\mathbf{M}$ over the $\mathbf{M}$–part and are empty over the w–part. The interpretations of the P_i's are empty over the $\mathbf{M}$–part, while for any $1 \leq k \leq lh(w)$, $P_i(M + k - 1)$ holds if and only if the k-th symbol in w is i. The interpretation of the arithmetic is, of course, standard.

For a given logic $\mathcal{L}$ and vocabulary σ, we associate with $\mathcal{L}$ the following *combined complexity class*:

$$CC_{\mathcal{L},\sigma} = \{(\mathbf{M} + \varphi) : \mathbf{M} \text{ is a } \sigma\text{–model}, \varphi \text{ is an } \mathcal{L}\text{–sentence of vocabulary } \sigma, \text{ and } M \models \varphi\}.$$

Observe that the complexity of $CC_{\mathcal{L},\sigma}$ is exactly the combined complexity of $\mathcal{L}$ over σ.

2.4 Finite order logic

In section 5, we concentrate on logics which are syntactically defined fragments of finite order logic $\mathcal{L}^\omega$. Finite order logic is somewhat like second order logic except that quantification over relations of arbitrary order is allowed. Since $\mathcal{L}^\omega$ is less commonly known than second order logic, a short introduction into the topic is necessary. For a detailed survey on finite order logic, refer to [L94].

The set **Typ** of *types* is defined inductively as the smallest set which satisfies: $\iota \in \mathbf{Typ}$, and for any $\tau_1, \ldots, \tau_k \in \mathbf{Typ}$, also $(\tau_1 \ldots \tau_k) \in \mathbf{Typ}$. The language of $\mathcal{L}^\omega$ has countably many variables for each $\tau \in \mathbf{Typ}$ (for example, x_i^τ for $i \in \omega$). The intended interpretation is that objects of type ι are individuals, i.e. elements of the universe of a given model, whereas objects of type $(\tau_1 \ldots \tau_k)$ are k–ary relations whose i–th argument is always an object of type τ_i.

The set of types is naturally stratified into orders, namely, $order(\iota) = 1$ and $order((\tau_1 \ldots \tau_k)) = max\{order(\tau_1), \ldots, order(\tau_k)\} + 1$. The order of a variable is the order of its type. By convention, first order variables are often denoted by lower-case letters from the end of the Latin alphabet, i.e. x, y, z etc., and second order variables are denoted by the capital letters P, R, S etc. We shall also sometimes use higher order variables, and denote them by bold capital letters $\mathbf{P}, \mathbf{R}, \mathbf{S}$ etc. (adding superscripts with the exact type of the variable and indices whenever necessary or convenient).

For a given vocabulary σ, the set T_σ of (first order) terms consists of variables of type ι and individual constants from σ, and the set F_σ^ω of finite order formulae is the smallest set containing:

- $P(t_1, \ldots, t_k)$, for any k–ary predicate from σ and any $t_1, \ldots, t_k \in T_\sigma$;
- $t = t'$, for any first order terms $t, t' \in T_\sigma$;
- $x_i^{(\overbrace{\iota \ldots \iota}^{k})}(t_1, \ldots, t_k)$, for any $i, k \in \omega$ and $t_1, \ldots, t_k \in T_\sigma$;
- $x_i^\tau(x_{i_1}^{\tau_1} \ldots x_{i_k}^{\tau_k})$, for any $\tau = (\tau_1 \ldots \tau_k)$, $i, i_1, \ldots, i_k \in \omega$;
- $\neg\varphi, (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \Rightarrow \psi)$ and $(\varphi \equiv \psi)$, for any $\varphi, \psi \in F_\sigma^\omega$;
- $\exists x_i^\tau \varphi, \forall x_i^\tau \varphi$, for any $\tau \in \mathbf{Typ}, i \in \omega$, and $\varphi \in F_\sigma^\omega$.

The notion of a model of vocabulary σ is standard. The semantics for $\mathcal{L}^\omega$ is defined, just as for first order logic, via valuations and the satisfaction relation. For a given model $\mathbf{M}$ with universe M, a valuation is a function **val** with domain $\mathbf{Typ} \times \omega$ which fulfils the condition that $\mathbf{val}(\tau, i)$ (interpreted as the value of the variable x_i^τ) belongs to M_τ (the universe of type τ). M_τ is defined inductively by: $M_\iota = M$, $M_{(\tau_1 \ldots \tau_k)} = \mathcal{P}(M_{\tau_1} \times \ldots \times M_{\tau_k})$. Satisfaction of formulae from F_σ^ω is defined in the natural inductive way.

The restriction of $\mathcal{L}^\omega$ to formulae whose variables are all of order $\leq d$ is called d–th order logic and denoted $\mathcal{L}^d$. It may be easily verified that $\mathcal{L}^1$ is just the standard first order logic (FO), and $\mathcal{L}^2$ is the standard second order logic (SO).

An $\mathcal{L}^\omega$–formula is in prenex normal form if all quantifiers are at the beginning of the formula, forming a *quantifier prefix* which precedes the quantifier free part of the formula. Every $\mathcal{L}^d$–sentence is equivalent to an $\mathcal{L}^d$–sentence in prenex normal form. We may also assume that the d–th order quantifiers (if present) precede all the remaining quantifiers in the prefix.

It is well known that FO– and SO–formulae in prenex normal form comprise hierarchies whose levels are denoted $\Sigma_m^0(\Pi_m^0)$ and $\Sigma_m^1(\Pi_m^1)$, respectively. The definition of these hierarchies can be extended to higher order logics. Let the class $\Sigma_0^d(= \Pi_0^d)$ consist of those $\mathcal{L}^{d+1}$–formulae in normal form in which no $(d+1)$–st order quantifiers appear, and let Σ_{m+1}^d consist of formulae of the form $\exists x_{i_1}^{\tau_1} \ldots \exists x_{i_k}^{\tau_k} \varphi$, where φ is Π_m^d and $order(\tau_i) = n+1$ for $i = 1, \ldots, k$. (with Π_{m+1}^d defined dually). Clearly, every $\mathcal{L}^{d+1}$–sentence is equivalent to a Σ_m^d sentence for some m.

By Fagin's famous theorem ([F74]), Σ_1^1 (the purely existential part of second order logic) captures the complexity class NP. Similarly, Σ_m^1 captures Σ_m^p, the m-th level of the polynomial hierarchy ([S77]). Thus, in general, a class of models is in the polynomial hierarchy (PH) if and only if it is second order definable.

This correspondence between the descriptive and computational complexity of classes of models can also be extended to higher orders. By induction on d, we define the d–fold exponential function: exp_0 is the identity function on the natural numbers and $\exp_{d+1}(n)$ is $2^{exp_d(n)}$. Let the complexity class d–$NEXPTIME$ be $\bigcup_{k\in\omega} NTIME(exp_d(n^k))$. Let the m–th level of the d–$NEXPTIME$ hierarchy of complexity classes be $\bigcup_{k\in\omega} \Sigma_m$–$TIME(exp_d(n^k))$, where Σ_m–$TIME(f)$ is the class of problems solvable in time bounded by f using a Σ_m–machime, i.e. an alternating Turing machine which starts in an existential state and may switch between existential and universal states at most $m-1$ times during a computation. Observe that 0–$NEXPTIME$ is simply NP, the 0–$NEXPTIME$ hierarchy is the polynomial hierarchy, and 1–$NEXPTIME$ is $NEXPTIME$. The 1–$NEXPTIME$ hierarchy is known as the (full) weak exponential hierarchy and is usually denoted by $EXPH$. Its levels are denoted by Σ_m^{exp}.

Let the class $ELEMENTARY$ consist of problems solvable using resources bounded by exp_d for some d (observe that it does not matter whether we take "resources" to mean "deterministic time" or e.g. "alternating space"). $ELEMENTARY$ is the union of the d–$NEXPTIME$ hierarchies.

We have the following proposition, which summarizes the relationships between higher order logics and complexity classes:

Proposition 2.6 *For any $n \geq 1$:*

(a) Σ_1^d captures $(d-1)$–$NEXPTIME$, and for any $m \geq 1$, Σ_m^d captures the m–th level of the $(d-1)$–$NEXPTIME$ hierarchy;

(b) $\mathcal{L}^{d+1}$ captures the $(d-1)$–$NEXPTIME$ hierarchy.

(c) $\mathcal{L}^{\omega}$ captures $ELEMENTARY$.

Remark. Proposition 2.6, which seems to be considered an element of the folklore, has had a strange history. Part (c) was shown by Bennett in his 1962 thesis ([Ben62]). As reported in [Bör84], the case $m = 1$ of part (a) was shown by Christen in his 1974 thesis ([C74]). Part (b) is stated without proof in theorem 50 of [L89] (and it should be noted that at least the case $d = 2$, stated separately as theorem 48 of [L89], is based on a mistaken definition of the "full exponential time hierarchy").

Note, incidentally, that proposition 2.6 follows easily from theorem 5.6[2].

[2] A characterization of the expressive power of the Σ_m^d classes in terms of oracle machines instead of alternating machines has recently been given in [HTT03].

3 Truth definitions vs. combined complexity

Let us focus on one of our goals: determining the relationships between FM–truth definitions for a logic $\mathcal{L}$ and the combined complexity of $\mathcal{L}$.

Observe the following evident difference between FM–truth and combined complexity. The combined complexity of $\mathcal{L}$ is, at least over a fixed vocabulary σ, a fixed level of complexity. On the other hand, the complexity of verifying whether an element $\ulcorner\varphi\urcorner$ of a σ–model $\mathbf{M}$ satisfies some FM–truth definition $Tr_{\mathcal{L},\sigma}(x)$ depends on the specific features of the formula $Tr_{\mathcal{L},\sigma}(x)$. This is due to the "asymptotic" character of FM–truth definitions. For example, it may be easy to check whether $\mathbf{M} \models \varphi$ actually holds, but considerably harder (even undecidable) to find out whether $\mathbf{M}$ is large enough to ensure that $Tr_{\mathcal{L},\sigma}(x)$ properly determines the truth of φ.

Despite this difference, it is easy to show that if a (reasonable) logic $\mathcal{L}'$ expresses the combined complexity class $CC_{\mathcal{L},\sigma}$, then $\mathcal{L}'$ also defines FM–truth for $\mathcal{L}$ over σ. To this end, define the *canonical truth–definition class* $TD_{\mathcal{L},\sigma}$:

$$TD_{\mathcal{L},\sigma} = \{(\mathbf{M}, \ulcorner\varphi\urcorner) : \mathbf{M} \text{ is a } \sigma\text{–model}, \varphi \text{ is an } \mathcal{L}\text{–sentence of vocabulary } \sigma, \ulcorner\varphi\urcorner \in M, \text{ and } M \models \varphi\}.$$

Theorem 3.1 *For any logic $\mathcal{L}$ and any vocabulary σ, the class $TD_{\mathcal{L},\sigma}$ is first order reducible to $CC_{\mathcal{L},\sigma}$. Hence, for any logic $\mathcal{L}'$ which defines $CC_{\mathcal{L},\sigma}$ and is closed under first order reductions, it holds that $\mathcal{L} \ll_\sigma \mathcal{L}'$.*

Proof. The reduction is to yield $(\mathbf{M} + w)$ on input $(\mathbf{M}, \ulcorner w\urcorner)$ (we reduce pairs $(\mathbf{M}, c)$ where c is not of the form $\ulcorner w\urcorner$ to some fixed model not in $CC_{\mathcal{L},\sigma}$). We take it to be binary: the universe of the model we build will consist of those pairs (b, n) (where $b, n < M$) for which either $b = 0$ (this will be the $\mathbf{M}$–part of $(\mathbf{M}+w)$) or $b = 1$ and $n < lh(w)$ (this will be the w–part). Defining $lh(w)$ in terms of $\ulcorner w\urcorner$ is no difficulty at all: in our example with the 16–letter alphabet (see section 2.2), $lh(w)$ is the greatest number k for which $BIT(\ulcorner w\urcorner, 4k)$ holds.

Also the interpretations of the σ^+–symbols in our model are easily first order definable. For example, $P_{0000}(0, n)$ never holds and $P_{0000}(1, n)$ holds if and only if the $(n+1)$–st character in w is 0000, in other words, if and only if $\bigwedge_{i=1}^{4}(\neg BIT(\ulcorner w\urcorner, 4lh(w) - 4n - i))$ is true. $\square$

Thus, a logic strong enough to capture the combined complexity of $\mathcal{L}$ (i.e. define $CC_{\mathcal{L},\sigma}$) is also strong enough to define FM–truth for $\mathcal{L}$. Is there any chance for a converse? An exact converse is clearly impossible as, for example, $CC_{FO,\sigma}$ is $PSPACE$–complete (for any σ), whereas $TD_{FO,\sigma}$

is easily seen to be in $DSPACE(log^2 n)$ (this is because if $\ulcorner\varphi\urcorner \in \mathbf{M}$ then the length of φ is logarithmic in comparison to the size of $\mathbf{M}$))[3]. But might things change if we take, say, an appropriately padded version of $CC_{\mathcal{L},\sigma}$ (e.g. add an exponentially long string of redundant zeroes to each input)?

Such a possibility must also be excluded, for the following reason. As already noted, the combined complexity of a logic depends in general both on its expressive power and its syntax. In the case of FM–truth definitions it is not so. It turns out that the difficulty of defining FM–truth for a logic depends only on the expressive power. The syntax, as long as it is decidable, does not matter:

Theorem 3.2 *Let $\mathcal{L}_1$, $\mathcal{L}_2$ be logics, and let $\mathcal{L}$ be a logic closed under first order quantification and under taking conjunctions and disjunctions with first order formulae. In that case, for any σ, if $\mathcal{L}_1 \ll_\sigma \mathcal{L}$ and $\mathcal{L}_2 \leq_\sigma \mathcal{L}_1$, then $\mathcal{L}_2 \ll_\sigma \mathcal{L}$.*

Proof. The relation "x_1 is an $\mathcal{L}_1$–sentence of vocabulary σ, x_2 is an $\mathcal{L}_2$–sentence of vocabulary σ, and x_1 is equivalent to x_2 in all σ–models" is co–RE and therefore FM–representable. Let it be FM–represented by the first order formula $eq(x_1, x_2)$. Let the $\mathcal{L}$–formula $Tr_{\mathcal{L}_1}(x_1)$ be an FM–truth definition for $\mathcal{L}_1$ over σ. Then the $\mathcal{L}$–formula

$$Tr_{\mathcal{L}_2}(x_2) := \exists x_1(eq(x_1, x_2) \land \forall y < x_2(\neg eq(y, x_2)) \land Tr_{L_1}(x_1))$$

is an FM–truth definition for $\mathcal{L}_2$ over σ.

Indeed, let a be (the Gödel number of) an $\mathcal{L}_2$–sentence of vocabulary σ. There exists a number b such that b is the smallest (Gödel number of) $\mathcal{L}_1$–sentence equivalent to a. For all sufficiently large σ–models $\mathbf{M}$, it holds that $\mathbf{M} \models eq(b, a) \land \forall y < b(\neg eq(y, a))$. Similarly, for all sufficiently large $\mathbf{M}$, we have $\mathbf{M} \models Tr_{\mathcal{L}_1}(b)$ if and only if b is true in $\mathbf{M}$. Thus, for all sufficiently large $\mathbf{M}$, $\mathbf{M} \models Tr_{\mathcal{L}_2}(a)$ if and only if a is true in $\mathbf{M}$. $\Box$

Basing on theorems 3.1 and 3.2, we may summarize the connection between FM–truth definitions and combined complexity in the following corollary:

Corollary 3.3 *Let $\mathcal{L}, \mathcal{L}'$ be logics such that $\mathcal{L}'$ is closed under first order reductions, and let σ be a vocabulary.*

[3] A Turing machine may check the truth value of an FO-sentence φ in a model $\mathbf{M}$ by successively writing down tuples of witnesses for the quantifiers in φ and verifying the quantifier free part of φ for each such tuple. Writing down a witness for one first order quantifier requires $O(logM)$ bits, and there are $O(lh(\varphi))$ quantifiers. But if $lh(\varphi)$ is $O(logM)$, then $O(logM \cdot lh(\varphi))$ is $O(log^2 M)$.

(a) If there exists any logic $\tilde{\mathcal{L}}$ satisfying $\tilde{\mathcal{L}} \equiv_\sigma \mathcal{L}$ for which $\mathcal{L}'$ defines $CC_{\tilde{\mathcal{L}},\sigma}$, then $\mathcal{L} \ll_\sigma \mathcal{L}'$.

(b) If $\mathcal{L} \ll_\sigma \mathcal{L}'$, then there exists a logic $\tilde{\mathcal{L}}$ such that $\mathcal{L}'$ defines $CC_{\tilde{\mathcal{L}},\sigma}$, and for any $\mathcal{L}$–definable class K of σ-models there is an $\tilde{\mathcal{L}}$–definable finite variant of K.

Proof. (a) This follows directly from theorems 3.1 and 3.2.
(b) Let $Tr_{\mathcal{L},\sigma}(x)$ be the FM–truth definition. Define the syntax of $\tilde{\mathcal{L}}$ to be the same as of $\mathcal{L}$. For an $\mathcal{L}$–sentence φ viewed as an $\tilde{\mathcal{L}}$–sentence, define $\mathbf{M} \models \varphi$ to hold if and only if $\ulcorner\varphi\urcorner \in \mathbf{M}$ and $\mathbf{M} \models Tr_{\mathcal{L},\sigma}(\ulcorner\varphi\urcorner)$. It is straightforward to verify that $CC_{\tilde{\mathcal{L}},\sigma}$ is first order reducible to the class $\{(\mathbf{M}, c) : \mathbf{M} \models Tr_{\mathcal{L},\sigma}(c)\}$. $\Box$

Part (b) of the corollary is clearly not satisfactory because the logic $\tilde{\mathcal{L}}$ we construct in the proof is very artificial and badly behaved: it is not even closed under finite variants. It would be interesting to see whether for natural logics $\mathcal{L}$, part (b) still holds if we require $\tilde{\mathcal{L}}$ to satisfy some minimal natural assumptions.

4 FM–truth definitions are easy

Theorem 3.2 — the fact that the difficulty of defining FM-truth for a logic does not depend on its syntax — has important consequences. In particular, we can use it to show that defining FM–truth is "not very hard": if a logic $\mathcal{L}$ captures some natural complexity class $\mathcal{C}$, then the complexity of evaluating an FM–truth definition for $\mathcal{L}$ may be just "a little bit" above $\mathcal{C}$. The proof is basically an application of the simple idea used to prove the deterministic version of the complexity hierarchy theorems.

Theorem 4.1 *Let σ be a vocabulary and let $\mathcal{L}$ be a logic such that model checking for $\mathcal{L}$ over σ is in $DSPACE(f)$ (resp. $DTIME(f), NTIME(f), \Sigma_m$–$TIME(f)$), where f is some space–constructible (resp. time–constructible) function. Let g be a space–constructible (resp. time–constructible) function such that $f = o(g)$ (resp. $f \cdot log\, f = o(g)$). If the logic $\mathcal{L}'$, closed under first order quantification, under taking conjunctions and disjunctions with first order formulae, and under substituting variables for individual constants, captures at least $DSPACE(g)$ (resp. $DTIME(g), NTIME(g), \Sigma_m$–$TIME(g)$), then $\mathcal{L} \ll_\sigma \mathcal{L}'$.*

Proof. We elaborate only on the $DSPACE$ case. By theorem 3.2, it is enough to find any logic $\tilde{\mathcal{L}}$ which captures at least $DSPACE(f)$ such that $\tilde{\mathcal{L}} \ll_\sigma \mathcal{L}'$.

Let the set of $\tilde{\mathcal{L}}$–sentences consist of (the codes of) all deterministic Turing machines which work within space bounded by f (to ensure decidability of syntax, we may assume that the set of $\tilde{\mathcal{L}}$–sentences consists of (the codes of) all DTMs equipped with some standard device which forces the machine to reject whenever it tries to use more space than is allowed by f; this is possible because of the space–constructibility of f). The semantics of $\tilde{\mathcal{L}}$ is defined by positing that for any model $\mathbf{M}$ and machine T, $\mathbf{M} \models T$ if and only if T accepts $\mathbf{M}$. Obviously, $\tilde{\mathcal{L}}$ captures at least $DSPACE(f)$.

Consider the Turing machine $T_{\tilde{\mathcal{L}}}$ which on input $(\mathbf{M}, n)$, where M is a σ–model, checks whether n is of the form $\ulcorner T \urcorner$ for some $\tilde{\mathcal{L}}$–machine T, and rejects if it is not. Otherwise, $T_{\tilde{\mathcal{L}}}$ simulates T on $\mathbf{M}$ and accepts if and only if T does. Additionally, $T_{\tilde{\mathcal{L}}}$ uses a device which forces it to stop and reject whenever it tries to use more space than $g(M)$ (again, this is possible as g is space–constructible).

Clearly, $T_{\tilde{\mathcal{L}}}$ is a $DSPACE(g)$–machine. Since $\mathcal{L}'$ captures at least $DSPACE(g)$, there exists an $\mathcal{L}'$–sentence $Tr_{\tilde{\mathcal{L}}}(c)$ in the vocabulary $\sigma + c$ (where c is a new individual constant) such that $(\mathbf{M}, n) \models Tr_{\tilde{\mathcal{L}}}(c)$ if and only if $T_{\tilde{\mathcal{L}}}$ accepts $(\mathbf{M}, n)$.

Moreover, for any $\tilde{\mathcal{L}}$–sentence T, the machine $T_{\tilde{\mathcal{L}}}$ will be able to complete its computation on $(\mathbf{M}, \ulcorner T \urcorner)$ for all sufficiently large $\mathbf{M}$ (as $f = o(g)$). It therefore follows that the $\mathcal{L}'$–formula $Tr_{\tilde{\mathcal{L}}}(x)$ is an FM–truth definition for $\tilde{\mathcal{L}}$ over σ.

The case of time–bounded machines is similar, except for the additional $log\, f$ factor in the assumptions, introduced in order to allow $T_{\tilde{\mathcal{L}}}$ to simulate $\tilde{\mathcal{L}}$–machines with an arbitrarily large tape alphabet and number of tapes (see e.g. [BDG95]). □

This is another difference between FM–truth definitions and combined complexity. As already noted in section 2.3, the combined complexity of a naturally defined logic is usually one exponential level higher than the complexity class captured by that logic. Evaluating an appropriately chosen FM–truth definition for that logic may be a task of much lower complexity.

It is fairly simple to account for this difference: the basic reason why FM–truth can be "easier" than combined complexity is that FM–truth definitions work "asymptotically". According to definition 2.3, an FM–truth definition for $\mathcal{L}$ is only required, for any given $\mathcal{L}$–sentence φ, to determine whether $\mathbf{M} \models \varphi$ for *almost all* models $\mathbf{M}$. If we allowed a "universal algorithm" for $\mathcal{L}$ (such as the one in the definition of combined complexity) to make a finite number of mistakes for each $\mathcal{L}$–sentence, then the complexity of such an algorithm could also be significantly lowered. This fact can be easily derived from theorem 4.1; we state it without proof:

Theorem 4.2 *Let σ be a vocabulary and let $\mathcal{L}$ be a logic such that model checking for $\mathcal{L}$ over σ is in $DSPACE(f)$ (resp. $DTIME(f), NTIME(f), \Sigma_m$–$TIME(f)$), where f is some space–constructible (resp. time–constructible) function. Let g be a space–constructible (resp. time–constructible) function such that $f = o(g)$ (resp. $f \cdot log\ f = o(g)$), Then there exists a $DSPACE(g)$ algorithm (resp. a $DTIME(g), NTIME(g), \Sigma_m$–$TIME(g)$ algorithm) which on input $(\mathbf{M}+\varphi)$ (where $\mathbf{M}$ is a σ–model and φ is an $\mathcal{L}$–sentence) answers whether $\mathbf{M} \models \varphi$ and which makes at most finitely many mistakes for each φ.*

5 Case study: higher order logics

We now apply the general results of the previous sections to a special case: we characterize the logics for which the prenex classes of higher order logics (the Σ^d_m classes, for any $d, m \geq 1$) define FM–truth.

The following is a simple observation:

Proposition 5.1 *Let σ be a vocabulary and let $\mathcal{L} \ll_\sigma \Sigma^d_m$. Then there exists a number k such that model checking for $\mathcal{L}$ over σ is in Σ_m–$TIME(exp_{d-1}(n^k))$.*

Proof. Let φ be an $\mathcal{L}$–sentence of vocabulary σ and let $Tr_{\mathcal{L},\sigma}(x)$ be the Σ^d_m FM–truth definition for $\mathcal{L}$. By (a) of proposition 2.6, there is some k and some Σ_m machine T which checks whether $(\mathbf{M}, \ulcorner w \urcorner) \models Tr_{\mathcal{L},\sigma}(c)$ in time bounded by $exp_{d-1}(n^k)$. But then the machine T^φ which, on input $\mathbf{M}$, writes down $\ulcorner\varphi\urcorner$ on its tape, simulates T on $(\mathbf{M}, \ulcorner\varphi\urcorner)$, and accepts if and only if T does, is also a Σ_m machine and works in time bounded by $exp_{d-1}(n^k)$. Moreover, T^φ determines the truth value of φ correctly in all sufficiently large models. □

So, for example, if FM–truth for $\mathcal{L}$ over σ can already be defined in Σ^1_1, then there is some k such that model checking for any $\mathcal{L}$–sentence over σ can be done in nondeterministic time n^k.

Since for any d and k it holds that $exp_d(n^k) = o(exp_d(n^{k+1}))$, theorem 4.1 immediately implies that the converse of proposition 5.1 is also true. We thus have:

Theorem 5.2 *Let $\mathcal{L}$ be a logic and let σ be a vocabulary. For any $d \geq 1$, it holds that $\mathcal{L} \ll_\sigma \Sigma^d_m$ if and only if there exists a number k such that model checking for $\mathcal{L}$ over σ is in Σ_m–$TIME(exp_{d-1}(n^k))$.*

Example. In [MP96], Makowsky and Pnueli use a diagonal method somewhat akin to FM–truth definitions to prove a hierarchy theorem for fragments of second order logic. They define $AA(k, m)$ as the class of those second order formulae in prenex normal form (here it is *not* required that the second order quantifiers precede the first order quantifiers) in which no relational variables of arity greater than k appear, and the number of alternating quantifier blocks in the prefix (counting both first and second order quantifiers) does not exceed m. It is shown that $AA(k, m)$ is less expressive than the whole second order logic (actually, it is shown that $AA(1, m) < AA(3, m+4)$ and $AA(k, m) < AA(k + 1, m + 4)$ for $k \geq 2$).

Observe that model checking for the "existential" fragment of $AA(k, m)$ is clearly in Σ_m–$TIME(n^k)$, and model checking for the "universal" fragment is in Π_m–$TIME(n^k)$ (where Π_m machines are defined dually to Σ_m machines). Together, model checking for $AA(k, m)$ is in Σ_{m+1}–$TIME(n^k)$, so for any vocabulary σ, $AA(k, m) \ll_\sigma \Sigma^1_{m+1}$ and hence $AA(k, m) < \Sigma^1_{m+1}$. Since over ordered models (or even over arbitrary models, by the normal form theorem of [EGG96]) $\Sigma^1_{m+1} \leq \bigcup_{l\in\omega} AA(l, m + 2)$, it follows that for any k, m there exists a c such that $AA(k, m) < AA(k + c, m + 2)$.

There are some obvious corollaries of theorem 5.2:

Corollary 5.3 *For all $d, m \geq 1$ and any vocabulary σ, $\Sigma^d_m \ll_\sigma \Sigma^{d+1}_m$.*

Corollary 5.4 *For any logic $\mathcal{L}$, any vocabulary σ, and any $d \geq 2$, it holds that $\mathcal{L} \ll_\sigma \mathcal{L}^d$ if and only if there exist m, k such that model checking for $\mathcal{L}$ over σ is in Σ_m–$TIME(exp_{d-1}(n^k))$. Similarly, $\mathcal{L} \ll_\sigma \mathcal{L}^\omega$ if and only if there exists d such that model checking for $\mathcal{L}$ over σ is in d–$NEXPTIME$.*

Together, theorem 5.2 and corollary 5.4 show that yet another natural logical property of languages (having an FM–truth definition in one of the basic syntactic fragments of finite order logic) has a simple characterization in terms of computational complexity.

5.1 The classes $[\Sigma^d_m]^{\leq k}$

In the remainder of this section we introduce and discuss, for each $m \geq 1$ and $d \geq 2$, certain syntactically defined subclasses of Σ^d_m. We obtain characterizations of the expressive power of these subclasses, from which it follows via theorem 4.1 that FM-truth for each of them is definable is Σ^d_m. We prove this last result directly, explaining how to explicitly write down appropriate FM–truth definitions.

Define the *basic arity* (*ba*) of a type τ in the following way. If $order(\tau) = 2$, then $ba(\tau) = arity(\tau)$. If $order(\tau) > 2$ and $\tau = (\tau_1, \dots, \tau_k)$, then $ba(\tau) = ba(\tau_{i_0})$, where τ_{i_0} has maximal basic arity among the τ_i of maximal order.

Definition 5.5 *For $d \geq 2$, $m, k \geq 1$, $[\Sigma^d_m]^{\leq k}$ is the class of Σ^d_m formulae in which the $(d+1)$–st and d–th order variables have basic arity at most k.*

Remark. Observe that in the definition of $[\Sigma^d_m]^{\leq k}$ we could require *all* the variables to have basic arity at most k. This would not change the expressive power of $[\Sigma^d_m]^{\leq k}$, as, for example, in sufficiently large models a second order relation R of some high arity l can be "coded" by a third order relation $\mathbf{R}^{((\iota\iota))}$ of basic arity 2. We can take the $\mathbf{R}$ such that any tuple $(x_1, \dots, x_l)$ is in R if and only if the relation X consisting exactly of the pairs $(1, x_1), \dots (l, x_l)$ is in $\mathbf{R}$ — and all relations in $\mathbf{R}$ correspond to l–tuples in this way. This coding may then be transferred to higher orders.

It is fairly straightforward to characterize the complexity classes captured by $[\Sigma^d_m]^{\leq k}$.

Theorem 5.6 $[\Sigma^d_m]^{\leq k}$ *captures* $\bigcup_{c \in \omega} \Sigma_m$*–*$TIME(exp_{d-1}(cn^k))$ *(for $d \geq 2$, $m, k \geq 1$).*

Proof. ($\subseteq$) Fix k. A simple proof by induction on d shows that for any d and any type τ such that $order(\tau) \leq d+1$, $ba(\tau) \leq k$, the number of objects of type τ over an n–element set is bounded by $exp_d(cn^k)$ for some c. Thus, objects of type τ may be represented as binary strings of length at most $exp_{d-1}(cn^k)$.

Now fix $d \geq 2, m$ and let φ be a $[\Sigma^d_m]^{\leq k}$ sentence (in prenex form, with the $(d+1)$–st order quantifiers in front). To check whether φ holds in a model $\mathbf{M}$, a Σ_m machine T^φ does the following. Starting in an existential state, it guesses relations corresponding to the variables in the initial block of existential quantifiers in φ (this amounts to writing down a fixed number of binary strings of length bounded by $exp_{d-1}(cM^k)$ for some c). T^φ then switches to a universal state and guesses relations corresponding to the first block of universal quantifiers, switches back to existential, and so on. Once the witnesses for all the $(d+1)$–st order quantifiers have been guessed, it remains to deterministically check the truth of φ for all possible choices of relations corresponding to the d–th and lower order quantifiers.

The number of relations over $\mathbf{M}$ corresponding to one such quantifier is bounded by $exp_{d-1}(cn^k)$ for some c (here we need the assumption that also the d–th order variables in φ have basic arity bounded by k), and since for $d \geq 2$, $exp_{d-1}(c_1 n^k) \cdot exp_{d-1}(c_2 n^k) \leq exp_{d-1}((c_1 + c_2)n^k)$, the number of all possible choices, and hence the running time, obeys the required time bound.

($\supseteq$) This is a standard variation on the theme of the proof of Fagin's theorem (see e.g. [I99]). Let T be a Σ_m machine which works in time bounded by $exp_{d-1}(cn^k)$. Let us list the possible contents of a tape cell during a run of T as $\gamma_1, \ldots, \gamma_l$ (the content of a cell is either a symbol from the machine alphabet, if the head is not on the cell, or an ordered pair ¡current state of T, machine alphabet symbol¿, if the head is on the cell).

For any given computation path of T, we construct an $exp_{d-1}(cn^k) \times exp_{d-1}(cn^k)$ matrix whose rows are indexed by points in time (t) and whose columns are indexed by units of T's tape space (s). The symbol γ_i appears in position (t, s) of the matrix if and only if γ_i is the content of cell s at time t along the given computation path. Observe that for an input of size n, we can let both t and s vary between 1 and $exp_{d-1}(cn^k)$, since T is $exp_{d-1}(cn^k)$ time–bounded and therefore has no chance to use more than $exp_{d-1}(cn^k)$ tape cells. This means that we can code both t and s by relations of type $\tau = \overbrace{(\ldots(}^{d-2}\overbrace{\underbrace{(\iota\ldots\iota)}_{k}\ldots\underbrace{(\iota\ldots\iota)}_{k}}^{c}\overbrace{)\ldots)}^{d-2}$, since there are exactly $exp_{d-1}(cn^k)$ objects of type τ over a universe of size n. Observe that $order(\tau) = d$, $ba(\tau) = k$.

We now introduce for each γ_i a relation variable $\mathbf{R}_i^{(\tau\tau)}$ for which, given a computation path of T, $\mathbf{R}_i(t, s)$ is to hold whenever γ_i is the symbol in position (t, s) of the matrix corresponding to this computation path (more precisely, we need m separate copies of the variable $\mathbf{R}_i$; which copy is used depends on how many alternations between existential and universal states of T have occured before time t). Note that $order(\mathbf{R}) = d + 1$, $ba(\mathbf{R}) = k$. The $[\Sigma^d_m]^{\leq k}$ sentence which defines the class of models accepted by T has m alternating blocks of $(d + 1)$–st order quantifiers for the variables $\mathbf{R}_i$, followed by a formula in which no quantifiers of order greater than d appear, and which states that the input is appropriate and that the computation proceeds according to the transition function of T. $\square$

Remark. Theorem 5.6 holds also over models without built–in arithmetic or even ordering, as each of the $[\Sigma^d_m]^{\leq k}$ classes can easily define an ordering (even for $k = 1$, as long as $d \geq 2$).

Example. The class $NETIME$ is captured by $[\Sigma^2_1]^{\leq 1}$.

Corollary 5.7 *For $d \geq 2$, $m, k \geq 1$, $[\Sigma^d_m]^{\leq k} < \Sigma^d_m$.*

Corollary 5.8 *For all $d \geq 2, m, k \geq 1$, and for any vocabulary σ, $[\Sigma^d_m]^{\leq k} \ll_\sigma \Sigma^d_m$.*

Corollary 5.8 follows trivially from theorems 4.1 and 5.6, but we will also give a more direct proof, showing how to write down the FM–truth definitions. One more technical definition will be helpful:

Definition 5.9 *A simple type is a type of the form $\tau = \overbrace{(\ldots(}^{d}\underbrace{\iota\ldots\iota}_{k}\overbrace{)\ldots)}^{d}$ for some $d, k \in \omega$.*

Let $s[\Sigma^d_m]^{\leq k}$ be $[\Sigma^d_m]^{\leq k}$ restricted to variables of simple types only. By an argument similar to the one in the proof of theorem 5.6, it can be shown that any class of models in Σ_m–$TIME(exp_{d-1}(n^k))$ can be defined using a $s[\Sigma^d_m]^{\leq k}$ sentence. Hence, it follows from theorem 5.6 and the hierarchy theorem that $[\Sigma^d_m]^{\leq k}$ is semantically strictly weaker than $s[\Sigma^d_m]^{\leq k+1}$.

Proof of corollary 5.8. It is enough to prove that for any k, Σ^d_m can define FM–truth for $s[\Sigma^d_m]^{\leq k}$. As an example, we show how to construct a Σ^2_2 FM–truth definition for $s[\Sigma^2_2]^{\leq k}$ (for any k and σ). Proofs for other Σ^d_m classes are analogous.

Observe that it is enough to write an FM–truth definition for the fragment of $s[\Sigma^2_2]^{\leq k}$ in which all second and third order variables have basic arity exactly k (there is an effective procedure which, given a $s[\Sigma^2_2]^{\leq k}$ sentence, outputs a logically equivalent $s[\Sigma^2_2]^{\leq k}$ sentence of the required form; the procedure consists in "adding superfluous arguments").

We treat first order valuations as unary functions on the universe. If $\ulcorner x \urcorner$ is the Gödel number of some first order variable x, and h is a unary function on M (where $\mathbf{M} \models$ "($\ulcorner x \urcorner$) is a first order variable"[4]), then the valuation h assigns to x the value $h(\ulcorner x \urcorner)$.

Similarly, we treat second order valuations for k–ary relational variables as $(k+1)$–ary relations: if $\ulcorner X \urcorner$ is the Gödel number of some k–ary relational variable X, W is a $(k+1)$–ary relation on M (where

[4]That is, $\ulcorner x \urcorner$ satisfies in $\mathbf{M}$ some fixed formula which FM–represents "being a first order variable". We say that "$\mathbf{M}$ recognizes $\ulcorner x \urcorner$ as a first order variable". Observe that it might happen that some given $\mathbf{M}$ is large enough to contain $\ulcorner x \urcorner$, but too small to recognize it as a first order variable. However, by the definition of "FM–represents", any first order variable will be recognized as such in sufficiently large models.

$\mathbf{M} \models$ "$\ulcorner X \urcorner$ is a variable of type $(\underbrace{\iota \ldots \iota}_{k})$"), and $a_1, \ldots, a_k$ are elements of M, then the tuple $(a_1, \ldots, a_k)$ is in the relation assigned to X by W iff $W(\ulcorner X \urcorner, a_1, \ldots, a_k)$ holds.

Valuations for third order variables of type $((\underbrace{\iota \ldots \iota}_{k}))$ are treated in a similar way, although we have to quantify them differently so as to keep our truth definition within Σ^2_2. The FM–truth definition now reads (x is the free first order variable; third order variables are in bold; phrases in quotation marks should be replaced by appropriate FM–representing formulae):

$$\exists x_1 \exists x_2 \exists x_3 \Big\langle (x = x_1 * x_2 * x_3)$$

$$\wedge \text{ "}x_1 \text{ is a string of existentially quantified variables of type } ((\underbrace{\iota \ldots \iota}_{k}))\text{"}$$

$$\wedge \text{ "}x_2 \text{ is a string of universally quantified variables of type } ((\underbrace{\iota \ldots \iota}_{k}))\text{"}$$

$$\wedge \text{ "}x_3 \text{ is a second order formula with all second order variables of type } (\underbrace{\iota \ldots \iota}_{k})$$

$$\text{and possibly containing third order free variables occurring in } x_1 \text{ or } x_2\text{"}$$

$$\wedge \; \exists \mathbf{X}_1 \forall \mathbf{X}_2 \forall \mathbf{T} \Big\{$$

$$\forall h \forall W \Big\langle \big[\forall \ulcorner \mathbf{S} \urcorner \forall \ulcorner R \urcorner \forall y (\text{"}\mathbf{S} \text{ is a variable occurring in } x_1\text{"}$$

$$\wedge \text{ "}R \text{ is a variable occurring in } x_3\text{"} \;\wedge\; y = \ulcorner \mathbf{S}(R) \urcorner$$

$$\Rightarrow (\mathbf{T}(y, h, W) \equiv \exists T$$

$$(\forall z_1 \ldots \forall z_k (T(z_1, \ldots, z_k) \equiv W(\ulcorner R \urcorner, z_1, \ldots, z_k)) \wedge \; \mathbf{X}_1(\ulcorner \mathbf{S} \urcorner, T)))) \big]$$

$$\wedge \; \big[\text{an analogous clause for } x_2 \text{ and } \mathbf{X}_2\big]$$

$$\wedge \; \big[\forall \ulcorner v_1 \urcorner \forall \ulcorner v_2 \urcorner \forall \ulcorner v_3 \urcorner \forall y (y = \ulcorner v_1 + v_2 = v_3 \urcorner$$

$$\Rightarrow (\mathbf{T}(y, h, W) \equiv h(\ulcorner v_1 \urcorner) + h(\ulcorner v_2 \urcorner) = h(\ulcorner v_3 \urcorner))) \big]$$

$$\wedge \; \big[\text{clauses for the remaining vocabulary symbols and logical connectives}\big]$$

$$\wedge \; \big[\forall \ulcorner v \urcorner \forall y \forall y_1 (\text{"}y \text{ is } \ulcorner \exists v \urcorner \text{ concatenated with } y_1\text{"})$$

$$\Rightarrow (\mathbf{T}(y, h, W) \equiv \exists h' (\forall z \neq \ulcorner v \urcorner (h'(z) = h(z)) \wedge \mathbf{T}(y, h', W)))) \big]$$

$$\wedge \; \big[\text{the clause for the second order quantifier}\big] \Big\rangle$$

$$\Rightarrow \forall h \forall W \, \mathbf{T}(x_3, h, W) \Big\} \Big\rangle .$$

Clearly, the above formula is equivalent to a Σ^2_2 formula.

In all sufficiently large models, a given $s[\Sigma^2_2]^{\leq k}$ sentence x will be recognized as such and correctly decomposed into the third order prefix (consisting of the existential part x_1 and universal part x_2) and the second order part x_3 (possibly containing third order variables which are quantified in the prefix). Now, x is true in a model if and only if there exists a valuation for the variables in x_1 (this valuation is represented by $\mathbf{X}_1$) such that for all valuations for the variables of x_2 (represented by $\mathbf{X}_2$), the second order part x_3 — with third order parameters interpreted according to $\mathbf{X}_1$ and $\mathbf{X}_2$ — is true.

The truth of x_3 is expressed using the third order variable $\mathbf{T}$. Call a third order relation $\mathbf{T}$ (of the appropriate type) *well-behaved* if $\mathbf{T}$ has the property that: for any first and second order valuations h and W and any SO formula y with all second order variables of arity $\leq k$ (but possibly with third order variables from among those in x_1 or x_2) recognized by the model as a formula of this type, $\mathbf{T}(y, h, W)$ holds if and only if y is true under the valuations $\mathbf{X}_1$, $\mathbf{X}_2$, h, and W. Our truth definition says that for any well-behaved $\mathbf{T}$ and any valuations h and W, $\mathbf{T}(x_3, h, W)$ holds, which in all sufficiently large models will be equivalent to the truth of x_3 (under the valuations given by $\mathbf{X}_1$ and $\mathbf{X}_2$).

A possible difficulty is we must exclude the danger of conflicting requirements being imposed on well-behaved relations. We have to avoid a situation in which some element of a model seems (according to the proper FM–representing formulae) to be e.g. both some formula y and its negation, which would require $\mathbf{T}(y, h, W)$ both to hold and not to hold for any well-behaved $\mathbf{T}$ and any h, W (thus, no well-behaved relations could exist, and any given $s[\Sigma^2_2]^{\leq k}$ sentence would satisfy the truth definition in the given model). This, however, is only an apparent difficulty. In fact, for any RE relation R we may find an FM–representing formula $\varphi_R(\mathbf{x})$ which is *never* satisfied by any tuple $\mathbf{a}$ unless $R(\mathbf{a})$ really holds [5]. □

Remark. Yet another proof of corollary 5.8 could be based on considering the combined complexity of $[\Sigma^d_m]^{\leq k}$, which is easily seen to be in the class captured by Σ^d_m (over any vocabulary).

[5]It is not necessarily true that $\varphi_R(\mathbf{x})$ is *always* satisfied by a tuple which belongs to R. Basically, $\varphi_R(\mathbf{x})$ says that there exists an accepting computation of the Turing machine appropriate for R on input $\mathbf{a}$. It may happen that $R(\mathbf{a})$ holds, but the given model is too small to contain the computation.

6 Corollaries

We close this paper with some corollaries, the first of which is related to the following well–known open problem of descriptive complexity theory (open problem 7.13 in [I99]):

(*) given a fixed vocabulary σ, does there exist a number k such that model checking for all FO–sentences over σ can be done in deterministic time n^k?

A version of (*) can also be formulated for nondeterministic instead of deterministic time. Here again the answer is unknown.

We observe that there exists a very simple complexity–theoretical sufficient condition for the answer to be positive.

Corollary 6.1 *Let f be any space–constructible function such that $log = o(f)$. Then:*

(a) if $DSPACE(f) \subseteq P$, then for any σ there exists k such that $FO \subseteq DTIME(n^k)$ over σ.

(b) if $DSPACE(f) \subseteq NP$, then for any σ there exists k such that $FO \subseteq NTIME(n^k)$ over σ.

Proof. Consider (a). Assume that f satisfies the conditions of the corollary and that $DSPACE(f) \subseteq P$. Fix σ. Model checking for FO is in $LOGSPACE$, so by theorem 4.1, there exists an FM–truth definition for FO over σ which can be evaluated in $DSPACE(f)$ and thus, by the assumption, also in P. Let the complexity of evaluating this FM–truth definition be $DTIME(n^k)$. Then model checking for FO over σ is also in $DTIME(n^k)$.

The proof for (b) is analogous. □

It seems quite probable that the answer to (*) is NO, especially in the deterministic case. Corollary 6.1 suggests that proving this should be very difficult, as it would immediately imply not only $P \subsetneq PSPACE$, but also the separation of P from any deterministic space class above $LOGSPACE$.

Corollary 6.1 is just one of many results which have a virtually identical proof. We give two further examples:

Corollary 6.2 *Corollary 6.1 holds also with "FO" replaced by "DTC" (deterministic transitive closure logic) throughout.*

Hence, if for some f satisfying the assumptions of the corollary $DSPACE(f) \subseteq P$, then for any σ there is k such that $LOGSPACE$ is *contained in $DTIME(n^k)$ over σ. In particular, LOGSPACE defined as a class of languages over $\{0,1\}^*$ is contained in $DTIME(n^k)$ for some k.*

Similarly for NP and $NTIME(n^k)$.

Corollary 6.3 *If there exists a time–constructible function f which eventually dominates every polynomial and for which $NTIME(f) \subseteq \Sigma^p_m$, then for any σ there is k such that NP is contained in Σ_m–$TIME(n^k)$ over σ. Similarly, if $NTIME(f) \subseteq PSPACE$, then for any σ there is k such that $NP \subseteq DSPACE(n^k)$ over σ.*

Our final corollary is similar but slightly different. Recall that the boolean hierarchy BH (over NP) consists of all problems which can be described as boolean combinations of NP problems. BH is contained in L^{NP}, the class of problems solvable in logarithmic space using an NP oracle.

Corollary 6.4 [6] *Let f be a time–constructible function which eventually dominates every polynomial. If $NTIME(f) \subseteq BH$, then $BH \subsetneq L^{NP}$.*

Proof. Let $FO(\Sigma^1_1)$ be the first order closure of existential second order logic. It follows from results in [G97] that $FO(\Sigma^1_1)$ captures L^{NP} over ordered models. If $NTIME(f) \subseteq BH$, then obviously also $NTIME(f) \subseteq L^{NP}$. Hence, by theorem 4.1, $FO(\Sigma^1_1)$ has an FM–truth definition for Σ^1_1 (over every vocabulary). Let $Tr(x)$ be such a definition (over some fixed vocabulary).

Let $Bool(\Sigma^1_1)$ be the class of boolean cobinations of Σ^1_1 sentences. Obviously, $Bool(\Sigma^1_1)$ captures BH.

Consider now the following formula $Tr_{bool}(x)$ (once again, phrases in quotation marks should be replaced by appropriate FM–representing formulae):

$$\exists \alpha(p_1,\ldots,p_l)\exists < y_1,\ldots,y_l >$$

$$\Big\langle \text{“}\alpha \text{ is a propositional formula with the variables } p_1,\ldots,p_l\text{”}$$

$$\wedge \text{ “}y_1,\ldots,y_l \text{ are } \Sigma^1_1 \text{ sentences”}$$

$$\wedge \text{ “}x \text{ is the result of substituting } y_1,\ldots,y_l \text{ for } p_1,\ldots,p_l \text{ in } \alpha\text{ ”}$$

$$\wedge \text{ “}\alpha \text{ is the smallest propositional formula from which}$$

$$x \text{ arises via such a substitution”}$$

$$\exists h : \{p_1,\ldots,p_l\} \longrightarrow \{0,1\}\big[\forall i < k(h(p_1) = 1 \equiv Tr(y_i))$$

$$\wedge \text{ “the value assigned to } \alpha \text{ by } h \text{ is 1”}\big]\Big\rangle.$$

[6] The proof of this corollary is inspired by Marcin Mostowski's ideas related to truth definitions in the standard model of arithmetic.

$Tr_{bool}(x)$ is an $FO(\Sigma^1_1)$ formula (in particular, the finite valuation h is a first order object). Furthermore, it is easy to see that $Tr_{bool}(x)$ is an FM–truth definition for $Bool(\Sigma^1_1)$. But this means that $BH = L^{NP}$ cannot hold. Otherwise, $Bool(\Sigma^1_1)$ would have equal expressive power to $FO(\Sigma^1_1)$. By theorem 3.2, this would imply that $FO(\Sigma^1_1)$ has an FM–truth definition for itself, which is impossible by the finite version of Tarski's theorem. □

Acknowledgement. The author would like to thank Marcin Mostowski for helpful comments and discussions.

References

[BDG95] J. L. Balcázar, J. Díaz, and J. Gabarró, **Structural Complexity I. Second, Revised Edition**, Springer–Verlag 1995.

[Ben62] J. Bennett, *On Spectra*, Ph.D. Thesis, Princeton University 1962.

[Bör84] E. Börger *Decision Problems in Predicate Logic*, in G. Lolli et al. (eds), **Logic Colloquium '82**, North–Holland 1984, pp. 263–301.

[C74] C. A. Christen *Spektren und Klassen Elementarer Funktionen*, Ph.D. Thesis, ETH Zürich 1974.

[EGG96] T. Eiter, G. Gottlob and Y. Gurevich, *Normal Forms for Second-Order Logic Over Finite Structures, and Classification of NP Optimization Problems*, in **Annals of Pure and Applied Logic** 78(1996), pp. 111-125.

[F74] R. Fagin, *Generalized First-Order Spectra and Polynomial-Time Recognizable Sets*, in **Complexity of Computation, SIAM-AMS Proceedings** 7(1974), pp. 43–73.

[G97] G. Gottlob, *Relativized Logspace and Generalized Quantifiers over Finite Ordered Structures*, **Journal of Symbolic Logic** 62(1997), pp. 545–574.

[GLV98] G. Gottlob, N. Leone, and H. Veith, *Succinctness as a Source of Complexity in Logical Formalisms*, **Annals of Pure and Applied Logic** 97(1999), pp. 231–260.

[HTT03] L. Hella and J. M. Turull Torres, *Expressing Database Queries with Higher Order Logics*, Technical Report 5/2003, Information Systems Department, Massey University 2003.

[I99] N. Immerman **Descriptive Complexity**, Springer–Verlag 1999.

[L89] D. Leivant, *Descriptive Characterizations of Computational Complexity*, in **Journal of Computer and System Sciences** 39(1987), pp. 51–83.

[L94] D. Leivant, *Higher Order Logic*, in D. M. Gabbay et al. (eds), **Handbook of Logic in Artificial Intelligence and Logic Programming**, vol. 2, Oxford University Press 1994, pp. 228–321.

[MM01] M. Mostowski, *On Representing Concepts in Finite Models*, in **Mathematical Logic Quarterly** 47(2001), pp. 513–523.

[MM0?] M. Mostowski, *On Representing Semantics in Finite Models*, to appear in **Proceedings of Contributed Papers, LMPhSc 99**.

[MP96] J. A. Makowsky and Y. B. Pnueli, *Arity and Alternation in Second Order Logic*, in **Annals of Pure and Applied Logic** 78(1996), pp. 189–202.

[S77] L. Stockmeyer, *The Polynomial-Time Hierarchy*, in **Theoretical Computer Science** 3(1977), pp. 1–22.

[T33] A. Tarski, **Pojęcie prawdy w językach nauk dedukcyjnych**, Warszawskie Towarzystwo Naukowe, Warszawa 1933. English translation of the German version: *The concept of Truth in Formalized Languages*, in A. Tarski, **Logic, Semantics, Metamathematics**, Oxford University Press 1956, pp. 152–278.

[V82] M. Vardi, *The Complexity Of Relational Query Languages*, in **Proc. 14th ACM Symp. on Theory of Computing** 1982, pp. 137–146.

[Z0?] K. Zdanowski *Arithmetics in Finite but Potentially Infinite Worlds*, Ph.D. Thesis, Warsaw University, in preparation.