Complexity, Decidability and Completeness

Douglas Cenzer
Department of Mathematics
University of Florida Gainesville, Fl 32611
e-mail: cenzer@math.ufl.edu
Jeffrey B. Remmel *
Department of Mathematics
University of California at San Diego
La Jolla, CA 92093
e-mail: jremmel@ucsd.edu

February 21, 2006

Abstract

We give resource bounded versions of the Completeness Theorem for propositional and predicate logic. For example, it is well known that every computable consistent propositional theory has a computable complete consistent extension. We show that, when length is measured relative to the binary representation of natural numbers and formulas, every polynomial time decidable propositional theory has an exponential time ($EXPTIME$) complete consistent extension whereas there is a nondeterministic polynomial time (NP) decidable theory which has no polynomial time complete consistent extension when length is measured relative to the binary representation of natural numbers and formulas. It is well known that a propositional theory is axiomatizable (respectively decidable) if and only if it may be represented as the set of infinite paths through a computable tree (respectively a computable tree with no dead ends). We show that any polynomial time decidable theory may be represented as the set of paths through a polynomial time decidable tree. On the other hand, the statement that every polynomial time decidable tree represents the set of complete consistent extensions of some theory which is polynomial time decidable, relative to the tally representation of natural numbers and formulas, is equivalent to $P = NP$. For predicate logic, we develop a complexity theoretic version of the Henkin construction to prove a complexity theoretic version of the Completeness Theorem. Our results imply that that any polynomial space decidable theory Δ possesses a polynomial space computable model which is exponential space decidable and thus Δ has an exponential space complete consistent extension. Similar results are obtained for other notions of complexity.

*Dept. of Commerce Agreement 70-NANB5H1164 and NSF grant DMS-9306427.

1 Introduction

Complexity theoretic or feasible model theory is the study of resource-bounded structures and isomorphisms and their relation to computable structures and computable isomorphisms. This subject has been developed during the 1990's by Cenzer, Nerode, Remmel and others. See the survey article [11] for an introduction. Complexity theoretic model theory is concerned with infinite models whose universe, functions, and relations are in some well known complexity class such as polynomial time, exponential time, polynomial space, etc. By far, the complexity class that has received the most attention is polynomial time. One immediate difference between computable model theory and complexity theoretic model theory is that it is not the case that all polynomial time structures are polynomial time equivalent. For example, there is no polynomial isomorphism f with a polynomial time inverse f^{-1} which maps the binary representation of the natural numbers $Bin(\omega) = \{0\} \cup \{1\}\{0,1\}^*$ onto the tally representation of the natural numbers $Tal(\omega) = \{1\}^*$. This is in contrast with computable model theory where all infinite computable sets are computably isomorphic so that one usually only considers computable structures whose universe is the set of natural numbers ω.

There are two basic types of questions which have been studied in polynomial time model theory. First, there is the basic existence problem, i.e. whether a given infinite computable structure $\mathcal{A}$ is isomorphic or computably isomorphic to a polynomial time model. That is, when we are given a class of structures $\mathcal{C}$ such as a linear orderings, Abelian groups, etc., the following natural questions arise.

(1) Is every computable structure in $\mathcal{C}$ isomorphic to a polynomial time structure?

(2) Is every computable structure in $\mathcal{C}$ computably isomorphic to a polynomial time structure?

For example, the authors showed in [3] that every computable relational structure is computably isomorphic to a polynomial time model and that the standard model of arithmetic $(\omega, +, -, \cdot, <, 2^x)$ with addition, subtraction, multiplication, order and the 1-place exponential function is isomorphic to a polynomial time model. The fundamental effective completeness theorem says that any decidable theory has a decidable model. It follows that any decidable relational theory has a polynomial time model. These results are examples of answers to questions (1) and (2) above. However, one can consider more refined existence questions. For example, we can ask whether a given computable structure $\mathcal{A}$ is isomorphic or computably isomorphic to a polynomial time model with a standard universe such as the binary representation of the natural numbers, $Bin(\omega)$, or the tally representation of the natural numbers, $Tal(\omega)$. That is, when we are given a class of structures $\mathcal{C}$, we can ask the following questions.

(3) Is every computable structure in $\mathcal{C}$ isomorphic to a polynomial time structure with universe $Bin(\omega)$ or $Tal(\omega)$?

(4) Is every computable structure in $\mathcal{C}$ computably isomorphic to a polynomial time structure with universe $Bin(\omega)$ or $Tal(\omega)$?

It is often the case that when one attempts to answer questions of type (3) and (4) that the contrasts between computable model theory and complexity theoretic model theory become more apparent. For example, Grigorieff [15] proved that every computable linear order is isomorphic to a linear time linear order which has universe $Bin(\omega)$. However Grigorieff's result can not be improved to the result that every computable linear order is computably isomorphic to a linear time linear order over $Bin(\omega)$. For example, Cenzer and Remmel [3] proved that for any infinite polynomial time set $A \subseteq \{0,1\}^*$, there exists a computable copy of the linear order $\omega + \omega^*$ which is not computably isomorphic to any polynomial time linear order which has universe A. Here $\omega + \omega^*$ is the order obtained by taking a copy of $\omega = \{0, 1, 2, \ldots\}$ under the usual ordering followed by a copy of the negative integers under the usual ordering.

The general problem of determining which computable models are isomorphic or computably isomorphic to feasible models has been studied by the authors in [3], [4], and [7]. For example, it was shown in [4] that any computable torsion Abelian group G is isomorphic to a polynomial time group A and that if the orders of the elements of G are bounded, then A may be taken to have a standard universe, i.e. either $Bin(\omega)$ or $Tal(\omega)$. It was also shown in [4] that there exists a computable torsion Abelian group which is not isomorphic, much less computably isomorphic, to any polynomial time (or even any primitive recursive) group with a standard universe. Feasible linear orderings were studied by Grigorieff [15], by Cenzer and Remmel [3], and by Remmel [25, 26]. Feasible vector spaces were studied by Nerode and Remmel in [20] and [22]. Feasible Boolean algebras were studied by Cenzer and Remmel in [3] and by Nerode and Remmel in [21]. Feasible permutation structures and feasible Abelian groups were studied by Cenzer and Remmel in [4] and [7]. By a *permutation structure* $\mathcal{A} = (A, f)$, we mean a set A together with a unary function f which maps A one-to-one and onto A.

General conditions were given in [10] which allow the construction of models with a standard universe such as $Tal(\omega)$ or $Bin(\omega)$ and these conditions were applied to graphs and to equivalence structures. An *equivalence structure* $\mathcal{A} = (A, R^{\mathcal{A}})$ consists of a set A together with an equivalence relation. For example, it was shown that any computable graph with all but finitely many vertices of finite degree is computably isomorphic to a polynomial time graph with standard universe. On the other hand, a computable graph was constructed with every vertex having either finite degree or finite co-degree (i.e. joined to all but finitely many vertices) which is not computably isomorphic to any polynomial time graph with a standard universe. An *equivalence structure* $\mathcal{A} = (A, R^{\mathcal{A}})$ consists of a set A together with an equivalence relation. It was also shown that any computable equivalence structure is computably isomorphic to a polynomial time structure with a standard universe.

The main goal of this paper is to study the relationship between theories whose decision problem lies in some standard complexity class between poly-

First order theory of	Upper Bound	Lower Bound
1-1 Unary Function	$DSPACE(2^{cn})$, $NTIME(2^{cn^2})$	$\notin NTIME(2^{c'n})$ (for some c')
One Successor	$DSPACE(n^2)$	$\notin NSPACE(c'n)$
$(\mathbb{N}, <)$	$DSPACE(n^2)$	$\notin NSPACE(c'n)$
$(\mathbb{Z}, +, \leq)$	$DSPACE(2^{2^{cn}})$	$\notin NTIME(2^{2^{c'n}})$ (for some c')

Table 1: Complexity of Certain Theories

nomial time and exponential space and the complexity of the models of that theory. For example, an obvious first question is to study the complexity of the basic Henkin construction for proving Gödel's Completeness Theorem. It is well known that the Henkin construction is effective in that it can be used to prove that any complete decidable theory Γ has decidable model. See the survey paper by Harizanov [16] for details. Here we say that a set Γ of formulas in some first order language is *decidable* if the set of consequences of Γ, $Cn(\Gamma) = \{\phi : \Gamma \vdash \phi\}$, is computable. Note that if Γ itself is computable, then the set of consequences of Γ is, in general, only computably enumerable (c.e.) and not necessarily computable. Similarly we say a model or structure $\mathcal{M}$ whose universe is a computable set $M \subset \{0,1\}^*$ is *decidable* if the complete theory of $\mathcal{M}$ is computable, i.e. the set of all formulas $\phi[a_1, \ldots, a_k]$ such that $a_1, \ldots, a_n \in M$ and $\mathcal{M} \models \phi[a_1, \ldots, a_k]$ is a computable set. We say that $\mathcal{M}$ is *computable* if the atomic diagram of $\mathcal{M}$ is computable, i.e. the set of all atomic formulas $\phi[a_1, \ldots, a_k]$ such that $a_1, \ldots, a_n \in M$ and $\mathcal{M} \models \phi[a_1, \ldots, a_k]$ is a computable set.

For predicate logic, we develop a complexity theoretic version of the Henkin construction. Our basic result on the complexity of the Henkin construction will imply that a polynomial space decidable theory will have an exponential space decidable model $\mathcal{M}$. Hence a polynomial space decidable theory always has an exponential space complete consistent extension. In addition, we show that $\mathcal{M}$ may be constructed to have a standard universe, $Bin(\omega)$ or $Tal(\omega)$, where the atomic diagram of $\mathcal{M}$ is polynomial space computable. In general, the complexity of the complete theory will be exponential over the complexity of the given theory. We shall also study the complexity of the omitting types theorem and the complexity of the completeness theorem for propositional logic.

The complexity of logical theories was studied from a different point of view in Ferrante's and Rackoff's book [14]. In [14], the goal was to study the complexity of well known theories such as the theory of the natural numbers N under the usual ordering $<$, $Th(N, <)$, or the theory of the natural numbers with successor, $Th(N, S)$. In particular, one wants to give upper and lower bounds on the complexity of these theories. Table 1 provides some of the basic results in the subject. A longer table is given in [14].

To understand this table, row two says that the theory of one successor is decidable in deterministic space n^2 but is not decidable in non-deterministic space $s(n)$ for some $s(n)$ such that $s(n)$ is $O(n)$ while row four says that the theory of the integers with addition and order $(\mathbb{Z}, +, \leq)$ is decidable in deterministic space $2^{2^{cn}}$ for some c, but there is a c' such that $(\mathbb{Z}, +, \leq)$ is not decidable in non-deterministic time $2^{2^{c'n}}$.

We note that there is often a lower limit on the complexity of non-trival propositional or predicate logic theories. To be more precise, in propositional logic, the set of consistent, or satisfiable, sentences is the classic NP complete set. Now a sentence ϕ is valid if and only if $\neg\phi$ is not satisfiable. Thus the smallest theory, the set of valid sentences is Co-NP complete. On the other hand, any complete propositional theory is determined by its underlying set of propositional variables. That is, let $V = \{A_0, A_1, \dots\}$ be a set of propositional variables, S be any subset of V and $\Gamma(S)$ be the consequences of $\{A_i : i \in S\} \cup \{\neg A_i : i \notin S\}$. Then S is computable from $\Gamma(S)$ in constant time. On the other hand, given any sentence ϕ containing variables $A_{i_1}, \dots, A_{i_k}$, we can decide whether $\phi \in \Gamma(S)$ by first making each A_t true if it is in S and false if not, and then evaluating ϕ. That is, $\phi \in \Gamma(S)$ if and only if the value of ϕ is true. Thus $\Gamma(S)$ is computable from S in linear time and linear space. Thus there are complete propositional theories in any of the standard complexity classes such as linear time, linear space, polynomial time, polynomial space, etc.

For predicate logic, the set of valid sentences is Σ^0_1-complete so that the set of satisfiable sentences is Π^0_1-complete. For most computable structures $\mathcal{M}$, the theory of $\mathcal{M}$ is $PSPACE$ hard by the following observation. Suppose that $\mathcal{M}$ is a computable model and that there is a formula $\phi(x_1, \dots, x_k)$ such that both ϕ and $\neg\phi$ are satisfiable in $\mathcal{M}$. Then the theory of M is $PSPACE$ hard. This follows from the fact that the decision problem, QBF, of the satisfiability of quantified Boolean formulas is $PSPACE$ hard. (See Papadimitriou [24].) QFB is the decision problem for the set of satisfiable quantified Boolean formulas of the form

$$\beta = (Q_1 p_1)(Q_2 p_2) \dots (Q_n p_n)\alpha$$

where α is a Boolean expression in $x_1, \dots, x_n$, each quantifier Q_i is either $\exists$ or $\forall$, and the sequence of quantifiers $Q_1, \dots Q_n$ alternates between $\exists$ and $\forall$. Now let $\phi(x_1, \dots, x_k)$ be given so that both ϕ and $\neg\phi$ are satisfiable in $\mathcal{M}$. Then β may be simulated in $\mathcal{M}$ by

$$Q_1 x_{1,1} \dots Q_1 x_{1,k} \dots Q_n x_{n,1} \dots Q_n x_{n,k} \alpha(\phi(x_{1,1}, \dots, x_{1,k}), \dots, \phi(x_{n,1}, \dots, x_{n,k}))$$

Thus unless $P = PSPACE$, no non-trivial polynomial time decidable theory can be complete.

There are nice examples of computable models $\mathcal{M}$ whose theories are in $PSPACE$. That is, consider structures $\mathcal{A} = (A, f)$ where $f : A \to A$ is unary function. Note that our structures will always include the equality relation. We say that $\mathcal{A}$ is an *injection structure* if f is one-to-one and we say that $\mathcal{A}$ is a *permutation structure* if f is a bijection from A onto A. As seen in the table above, the complexity of the general theory of injection structures is in nondeterministic exponential space and is not NP. Cenzer and Remmel also studied *injection structures* in [4, 7]. Before we can give some examples of computable injections and/or permutation structures whose theories have low complexity, we need to establish some notation. The *orbit* $\mathcal{O}(a)$ of an element a of (A, f) is defined to be

$$\mathcal{O}(a) = \{b \in A : (\exists n \in \omega)(f^n(a) = b \vee f^n(b) = a)\}.$$

There are two types of infinite orbits, one of type ω which is isomorphic to (ω, S) and the other of type $\mathbb{Z}$ which is isomorphic to $(\mathbb{Z}, S)$. The *order* $|a|$ of an element $a \in A$ is $card(\mathcal{O}(a))$ if $\mathcal{O}(a)$ is finite, is ω if $\mathcal{O}$ under f is isomorphic to (ω, S), and is $\mathbb{Z}$ if $\mathcal{O}$ under f is isomorphic to $(\mathbb{Z}, S)$. The *full spectrum* of (A, f) consists of all pairs $(0, n)$ such that there are at least $n+1$ orbits of type ω, $(1, n)$ such that there are at least $n+1$ orbits of type $\mathbb{Z}$, and (q, n) such that $q > 1$ and there are at least $n+1$ orbits of size $q-1$ in (A, f^A).

We can use Ehrenfeucht-Fraisse games to show that for certain permutation structures $\mathcal{A} = (A, f)$, $Th(\mathcal{A})$ is in $PSPACE$. Recall that two structures are n-equivalent if they satisfy the same sentences of quantifier rank $\leq n$. Given two structures $\mathcal{A}$ and $\mathcal{B}$, the Ehrenfeucht-Fraisse game $G_n(\mathcal{A}, \mathcal{B})$ of length n has two players, the "duplicator" and the "spoiler". At turn i of the game, the spoiler selects an element $a_i \in A$ or $b_i \in B$ from one of the two structures and the duplicator then selects an element from the other structure. The duplicator wins a play of this game if for the chosen elements, the substructure $\{a_1, \dots, a_n\}$ of $\mathcal{A}$ is isomorphic to the substructure $\{b_1, \dots, b_n\}$. The classic theorem of Ehrenfeucht shows that two structures are n-equivalent if and only if the duplicator has a winning strategy for the Ehrenfeucht game $G_n(\mathcal{A}, \mathcal{B})$. See [12] for a detailed exposition on Ehrenfeucht-Fraisse games.

Now let $\mathcal{M} = (M, f)$ be the permutation structure which consists of infinitely many orbits, each of finite size k. It is not hard to see that, for each n, the structure (M, f) is n-equivalent to the finite structure (M_n, g), consisting of n orbits of size k. That is, given that $\{a_1, \dots, a_{i-1}\}$ is isomorphic to $\{b_1, \dots, b_{i-1}\}$, we note that at most $i-1$ orbits of M_n have elements among $\{b_1, \dots, b_{i-1}\}$. Now given a new element $a_i \in M$, there are two cases. First, we might have $a_i = g^t(a_j)$ for some $t < k$ and some $j < i$. Then the duplicator just selects $b_i = g^t(b_j)$ to extend the isomorphism. Second, we might have a_i in a different orbit from each of $a_1, \dots, a_{i-1}$. Then the duplicator simply selects b_i from an orbit which does not include any of $b_1, \dots, b_{i-1}$. Now suppose we are given a sentence ϕ of length n and hence of quantifier depth $\leq n$. To check whether $\mathcal{M} \models \phi$, it suffices to check whether $(M_n, g) \models \phi$. Since M_n has only nk elements and ϕ has length n, this can easily be done in polynomial space $(nk)^r$ for some r. This shows that $Th(\mathcal{M})$ is in $PSPACE$. It is not hard to generalize this argument to permutation structures which consists of all finite orbits of finitely many different sizes.

It was shown in [7] that any computable model of a 1-1 unary function is isomorphic to a polynomial time model but not necessarily to a polynomial time model with standard universe. Various conditions under which a standard universe may be obtained were given, including when the tally representation of the spectrum is in P. Now the spectrum of (A, f) is computable in polynomial time from the theory, since for example, $(4, 0)$ is in the spectrum if and only if

$$(\exists x)[fff(x) = x \ \& \ ff(x) \neq x].$$

It follows first that any complete, decidable theory of a 1:1 unary function has a polynomial time model and second that any complete polynomial time decidable theory has a polynomial time model with standard universe. The main theorem of this paper concerning the complexity of the Henkin construction will give a more general result in the special case of polynomial time decidable theories. That is, we shall show that every polynomial time decidable theory has a polynomial time model with standard universe which is also exponential time decidable.

The outline of this paper is as follows. In section two, we shall first establish our basic notation for complexity classes. Then we shall study the complexity of theories and models for propositional logic. In particular, we will show that any theory which is polynomial time decidable in tally has a polynomial time complete consistent extension while there exists a theory which is NP decidable in binary but has no polynomial time decidable complete consistent extension. We also consider the representation of the set of complete consistent extensions of a theory by the Π^0_1 class of infinite paths through a tree. Here we show that the following is equivalent to $P = NP$: Every polynomial time decidable tree represents the set of complete consistent extensions of some theory which is polynomial time decidable in tally. In section three, we shall prove our main theorem that any polynomial space decidable theory has an exponential space decidable model with standard universe which is also polynomial space computable. Similar results are given for other notions of complexity. Our notation will follow that of [10]. For more on the theory of computability, see Odifreddi [23], Soare [27], for more on complexity theory, see Hopcroft and Ullman [17]. For more on computable model theory, see Harizanov [16].

2 Propositional Theories

2.1 Basic complexity definitions

We start this section by giving the basic definitions from complexity theory which will be needed for the rest of the paper.

Let Σ be a finite alphabet. Then Σ^* denotes the set of finite strings of letters from Σ and Σ^ω denotes the set of infinite strings of letters from Σ where $\omega = \{0, 1, 2, \ldots\}$ is the set of natural numbers. For any natural number $n \neq 0$, $tal(n) = 1^n$ is the tally representation of n and $bin(n) = i_e \ldots i_1 i_0 \in \{0, 1\}^*$ is the binary representation of n if $n = i_0 + 2 \cdot i_1 + \cdots + 2^e \cdot i_e$ and $i_e \neq 0$. In general, the k-ary representation $b_k(n) = i_e \ldots i_1 i_0$ if $n = i_0 + i_1 \cdot k + \cdots i_e \cdot k^e$ and $i_e \neq 0$. We let $tal(0) = bin(0) = b_k(0) = 0$. Then we let $Tal(\omega) = \{tal(n) : n \in \omega\}$, $Bin(\omega) = \{bin(n) : n \in \omega\}$ and, for each $k \geq 2$, $B_k(\omega) = \{b_k(n) : n \in \omega\}$.

For a string $\sigma = (\sigma(0), \sigma(1), \ldots, \sigma(n-1))$, $|\sigma|$ denotes the length n of σ. The empty string has length 0 and will be denoted by $\emptyset$. A constant string σ of length n will be denoted by k^n. For $m < |\sigma|$, $\sigma\lceil m$ is the string $(\sigma(0), \ldots, \sigma(m-1))$; σ is an *initial segment* of τ (written $\sigma \prec \tau$) if $\sigma = \tau\lceil m$ for some m. The *concatenation* $\sigma^\frown\tau$ (or sometimes just $\sigma\tau$) is defined by

$$\sigma^\frown\tau = (\sigma(0), \sigma(1), \ldots, \sigma(m-1), \tau(0), \tau(1), \ldots, \tau(n-1)),$$

where $|\sigma| = m$ and $|\tau| = n$; in particular we write $\sigma^\frown a$ for $\sigma^\frown(a)$ and $a^\frown\sigma$ for $(a)^\frown\sigma$.

Our basic computation model is the standard non-deterministic multitape Turing machine of Hopcroft and Ullman [15]. Note that there are different heads on each tape and that the heads are allowed to move independently. This implies that a string σ can be copied in linear time. An oracle machine is a multitape Turing machine M with a distinguished work tape, a query tape, and three distinguished states QUERY, YES, and NO. At some step of a computation on an input string σ, M may transfer into the state QUERY. In state QUERY, M transfers into the state YES if the string currently appearing on the query tape is in an oracle set A. Otherwise, M transfers into the state NO. In either case, the query tape is instantly erased. The set of strings accepted by M relative to the oracle set A is $L(M, A) = \{\sigma|$ there is an accepting computation of M on input σ when the oracle set is $A\}$. If $A = \emptyset$, we write $L(M)$ instead of $L(M, \emptyset)$.

Let $t(n)$ be a function on natural numbers. A Turing machine M is said to be $t(n)$ *time bounded* if each computation of M on inputs of length n where $n \geq 2$ requires at most $t(n)$ steps. A function $f(x)$ on strings is said to be in $DTIME(t)$ (respectively, $NTIME(t)$) if there is a $t(n)$-time bounded deterministic (resp. non-deterministic) Turing machine M which computes $f(x)$. For a function f of several variables, we let the length of $(x_1, \ldots, x_n)$ be $|x_1| + \cdots + |x_n|$. A set of strings or a relation on strings is in $DTIME(t)$ if its characteristic function is in $DTIME(t)$. A Turing machine M is said to be $t(n)$ *space bounded* if each computation of M on inputs of length n where $n \geq 2$ the work space required to carry out the computation is bounded by $t(n)$. A function $f(x)$ on strings is said to be in $DSPACE(t)$ (respectively, $NSPACE(t)$) if there is a $t(n)$-space bounded deterministic Turing machine M which computes $f(x)$. A set of strings or a relation on strings is in $DSPACE(t)$ (resp. $NSPACE(t)$ if its characteristic function is in $DSPACE(t)$. For a family $\mathcal{T}$ of functions, $DTIME(\mathcal{T}) = \bigcup_{f \in \mathcal{T}} DTIME(f)$ and similarly for $NTIME$, $DSPACE$ and $NSPACE$. The family $f(O(n))$ denotes $\{f(cn) : c \in \omega\}$. In particular, $LIN = DTIME(O(log(n)))$, $DEXT = DTIME(2^{O(n)})$ (exponential time), and $EXPSPACE = DSPACE(2^{O(n)})$. In addition to the standard complexity classes P, NP and $PSPACE$, we will also consider double exponential time and space as well as the interesting class $DTIME(n(log\ n)^{O(1)})$, which was studied by R. Brent in [1], plus the following:.

$LOG = \bigcup_{c \geq 1} DSPACE(c \cdot log_2(n))$,

$EXPTIME = \bigcup_{c\geq 0} DTIME(2^{n^c})$, and

$NEXPTIME = \bigcup_{c\geq 0} NTIME(2^{n^c})$. $DOUBEXT = \bigcup_{c\geq 0} DTIME(2^{2^{c\cdot n}})$,

$DOUBEXPSPACE = \bigcup_{c\geq 0} DSPACE(2^{2^{c\cdot n}})$,

We fix enumerations $\{P_i\}_{i\in N}$ and $\{N_i\}_{i\in N}$ of the polynomial time bounded deterministic oracle Turing machines and the polynomial time bounded non-deterministic oracle Turing machines respectively. We may assume that $p_i(n) = max(2,n)^i$ is a strict upper bound on the length of any computation by P_i or N_i with any oracle X on inputs of length n. P_i^X and N_i^X denote the oracle Turing machine using oracle X.

For $A,\ B \subset \Sigma^*$, we shall write $A \leq_m^P B$ if there is a polynomial-time function f such that for all $x \in \Sigma^*$, $x \in A$ iff $f(x) \in B$. We shall write $A \leq_T^P B$ if A is polynomial time Turing reducible to B. For r equal to m or T, we write $A \equiv_r^P B$ if $A \leq_r^P B$ and $B \leq_r^P A$ and we write $A \,|_r^P B$ if not $A \leq_r^P B$ and not $B \leq_r^P A$.

2.2 Complexity of propositional theories

We start this subsection with some definitions and background. Let the propositional language $\mathcal{L}$ have propositional letters $A_0, A_1, \ldots$ and connectives $\neg$, $\vee$ and $\wedge$ and let $Sent(\mathcal{L})$ be the set of sentences in this language. For any subset Δ of $Sent(\mathcal{L})$, let $Cn(\Delta)$ be the set of consequences of Δ, that is, $Cn(\Delta) = \{\phi : \Delta \vdash \phi\}$. Let $SAT(\Delta) = \{\phi : \Delta \cup \{\phi\}$ is consistent (or satisfiable)$\}$. A subset Γ of $Sent(\mathcal{L})$ is said to be a *theory* if it is closed under consequences, that is, if $Cn(\Gamma) = \Gamma$. A theory Γ is said to be *axiomatizable* if there is a computably enumerable set Δ such that $\Gamma = Cn(\Delta)$. A theory Γ is said to be *decidable* if there is a computable algorithm for deciding whether a given sentence ϕ is in Γ.

Shoenfield observed in [28] that for any axiomatizable predicate logic or propositional logic theory Γ, the set $E(\Gamma)$ of complete consistent extensions of Γ may be represented as a Π_1^0 class, that is, as the set of infinite paths through a computable tree. For a decidable theory, this tree may be taken to have no dead ends. In fact, Ehrenfeucht [13] proved that any Π_1^0 class represents an axiomatizable theory. For a tree without dead ends, the theory may be taken to be decidable. This representation is central to the discussion of the complexity of theories. See [11] for a more complete discussion of the relationship between axiomatizable (decidable) theories and Π_1^0 classes. Next we shall consider complexity theoretic versions of these results.

It is first necessary to define the length $|\phi|$ of a formula ϕ. Suppose that the underlying set of propositional letters in our propositional language is $\{A_0, A_1, \ldots\}$. In the standard or binary representation of a sentence ϕ, the numeral i in a propositional letter A_i is written in binary representation $bin(i)$ so that the length $|A_i|$ in binary is $1 + |bin(i)|$. That is, $|bin(A_i)| = r + 2$ when

$2^r \leq i < 2^{r+1}$. In the tally representation, the numeral i is written as 1^i so that $|tal(A_i)| = i+1$. A complete consistent theory Γ is represented by a subset of ω, $S(\Gamma) = \{i : A_i \in \Delta\}$, or, equivalently, by the characteristic function in $\{0,1\}^\omega$ of $S(\Gamma)$. The set of all complete consistent extensions of a consistent set Δ of sentences is denoted as $CC(\Delta)$. We shall let a finite sequence $\sigma \in \{0,1\}^n$ represent the sentence $B(\sigma) = B_0 \wedge B_1 \wedge \ldots B_n$, where $B_i = A_i$ if $\sigma(i) = 1$ and $B_i = \neg A_i$ if $\sigma(i) = 0$.

Lemma 2.1 *$tal(B(\sigma))$ has length $O(n^2)$ and may be computed in time $O(n^2)$ and $Bin(B(\sigma))$ has length $O(n \cdot log\ n)$ and may be computed in time $O(n\ log\ n)$.*

A set Δ of sentences is said to be *P-decidable* in binary (in tally) if there is a polynomial time Turing machine which given as input the binary (tally) representation of a formula ϕ, computes 1 if $\Delta \vdash \phi$ and computes 0 otherwise. We say that Δ is *weakly P-decidable in binary* (in tally) if there is a polynomial time Turing machine which given as input the binary (tally) representation of a conjunction ϕ of literals, computes 1 if $\phi \in SAT(\Delta)$ and computes 0 otherwise. One can define the notion of Δ being (weakly) $\mathcal{C}$-decidable in binary or tally for any complexity class $\mathcal{C}$ in a similar manner. A theory Γ is said to be P-axiomatizable if it possesses a polynomial time set Δ of axioms such that $\Gamma = Cn(\Delta)$. Again similar definitions apply to other notions of complexity.

For a tree $T \subset \{0,1\}^{<\omega}$, we say that T is a *P-tree* if the set $\{\sigma \in \{0,1\}^n : \sigma \in T\}$ is a polynomial time set. We will say that T is *P-decidable* if T is a P-tree and the set of dead ends of T is also in P (so that there is a P-tree S with no dead ends such that $[S] = [T]$). Similar definitions apply to other notions of complexity. The complexity of various aspects of trees was studied by the authors in [6].

We say that the tree T represents $CC(\Delta)$, the set of complete consistent extensions of Δ, if the set $[T]$ of infinite paths through T equals the family of sets $S \subseteq \{A_0, A_1, \ldots\}$ such that $\Gamma(S)$ is a complete consistent extension of Δ. We observe that there is a canonical tree T which represents $CC(\Delta)$ where $\sigma \in T \iff B(\sigma) \in SAT(\Delta)$.

Theorem 2.2 *Let Δ be a propositional theory.*

(i) If Δ is weakly $DTIME(n\ log(n)^{O(1)})$ decidable in binary, then $CC(\Delta)$ may be represented as the set of paths through a tree in $DTIME(n\ log(n)^{O(1)})$.

(ii) If Δ is weakly P-decidable (respectively $PSPACE$) in either binary or tally, then $CC(\Delta)$ may be represented as the set of paths through a P-tree (resp. $PSPACE$-tree).

(iii) If Δ is weakly $DEXT$-decidable (respectively, $EXPSPACE$-decidable) in tally or binary, then $CC(\Delta)$ may be represented as the set of paths through an $EXPTIME$-tree (resp. $\bigcup_{k \in \omega} DSPACE(2^{n^k})$-tree).

Proof: In each case, we shall let T be the canonical tree which represents $CC(\Delta)$. That is, $\sigma \in T \iff B(\sigma) \in SAT(\Delta)$.

(i) Suppose that Δ is weakly $DTIME(n\ log(n)^{O(1)})$ decidable in binary. By Lemma 2.1, we can compute $bin(B(\sigma))$ from σ in time $O(n\ log\ n)$, so that T is in $DTIME(n\ log(n)^{O(1)})$.

(ii) It easily follows from Lemma 2.1 that we can compute $bin(B(\sigma))$ and $tal(B(\sigma))$ in polynomial time and space from σ. Thus if Δ is weakly P-decidable (weakly $PSPACE$-decidable), then T is a P-tree ($PSPACE$-tree).

(iii)If Δ is weakly $DEXT$-decidable in tally ($EXPSACE$-decidable), it will require on the order of $2^{|\sigma|^2}$ time (space) to determine if $B(\sigma) \in SAT(\Delta)$ so that T is an $EXPTIME$-tree ($\bigcup_{k\in\omega} DSPACE(2^{n^k})$-tree). Similarly if Δ is weakly $DEXT$-decidable in binary ($EXPSACE$-decidable), it will require on the order of $2^{|\sigma| log(|\sigma|)}$ time (space) to determine if $B(\sigma) \in SAT(\Delta)$ so that again T is an $EXPTIME$-tree ($\bigcup_{k\in\omega} DSPACE(2^{n^k})$-tree). $\Box$

The corresponding result for axiomatizable theories does not require any restriction on the complexity of the set of axioms. In fact, our next result strengthens Theorem 4.1 of [5] which showed that any Π^0_1 class may be represented as the set of paths through a polynomial time tree.

A computable function f is said to be *time constructible* if and only if there is a Turing machine which on every input of size n halts in exactly $f(n)$ steps. In particular, the functions $log_2^k(n)$ are time constructible for $k \geq 1$ where we define $log_2^k(n)$ by induction as $log^1(n) = log(n)$ and for $k > 1$, $log_2^k(n) = log_2(log_2^{k-1}(n))$.

Theorem 2.3 *Let f be any time constructible function which is nondecreasing and unbounded. If a propositional theory Γ has a computably enumerable set of axioms, then it has a $DTIME(O(f))$ set of axioms and may be represented as the set of paths through a $DTIME(O(f))$-tree.*

Proof: First we need to construct a computable nondecreasing unbounded function g such that

1. $g(n) \leq f(n)^{1/4}$ for all n and

2. $g \in DTIME(O(f))$ relative to the tally representation.

It is easy to compute such a g. That is, suppose that $m \leq f(n)^{\frac{1}{10}}$, then we can compute the tally representation of m^4 in time cm^4 for some constant c. It follows that we can compute m^4 for $j = 0, 1, \ldots, m$, in time

$$\sum_{j=0}^{m} cj^4 \leq cm^5 \leq cf(n)^{1/2}.$$

Now for sufficiently large n, $cf(n)^{1/2} \leq f(n)$. Thus to compute $g(n)$, we start computing the values of j^4 for $j = 0, 1, \ldots$ until we have used $f(n)$ steps. $g(n)$ is the largest m such that we have finished the computation of j^4 for $j = 0, 1, \ldots, m$. Note that g is clearly nondecreasing and is unbounded since f

is unbounded. Since $g(n)$ is computed in time $f(n)$, we must have $g(n)^4 \leq f(n)$ so that (1) and (2) hold.

Since Γ has a c.e. set of axioms, it follows that Γ itself is computably enumerable. For each t, let Γ_t be the set of sentences enumerated into Γ by stage t. Now given a finite path $\sigma \in \{0,1\}^n$, put $\sigma \in T$ if and only if, for all $m < g(n)$, $\neg B(\sigma\lceil m) \notin \Gamma_{g(n)}$. Then for each $m < g(n)$, $B(\sigma\lceil m)$ may be computed in time $O(m^2)$ and has length $O(m^2)$. Since $\Gamma_{g(n)}$ has at most $g(n)$ strings, we need only make $g(n)$ comparisons to see if $\neg B(\sigma\lceil m) \notin \Gamma_{g(n)}$. Since each comparison takes $O(n^2)$ time, it take $O(m^2)g(n))$ time to decide if $\neg B(\sigma\lceil m) \notin \Gamma_{g(n)}$. Thus the total time needed to test $\sigma \in T$ is less than or equal to $\sum_{m=0}^{g(n)-1} cm^2 g(n) \leq cg(n)^4 \leq cf(n)$ for some constant c.

We claim that $x \in [T]$ if and only if x represents a complete consistent extension of Γ. Suppose first that $x \notin T$ and recall that x represents the theory $\Gamma(x)$ which contains A_n whenever $x(n) = 1$ and contains $\neg A_n$ whenever $x(n) = 0$. That is, $\Gamma(x)$ contains $B(x\lceil n)$ for all n. Since $x \notin T$, there exists m such that $\sigma = x\lceil m \notin T$. Then by the definition of T, $\neg B(\sigma\lceil n) \in \Gamma$ for some $n < m$ which shows that $\Gamma(x)$ is not consistent with Γ. Next suppose that $\Gamma(x)$ is not consistent with Γ. Then for some n, $\neg B(x\lceil n) \in \Gamma$. Thus there is an s such that $\neg B(x\lceil n) \in \Gamma_s$. Now let $r = max\{n, s\}$ and choose m large enough so that $g(m) \geq r$. It follows from our definitions that $x\lceil m \notin T$.

Finally, we note that Γ has a $DTIME(O(f(n)))$ set of axioms Δ where Δ is simply the set of $\neg B(\sigma)$ such that $\sigma \notin T$. $\square$

Thus for any k and any axiomatizable theory Γ, Γ has a $DTIME(O(log^k(n)))$ set of axioms and may be represented as the set of paths through a tree which lies in $DTIME(O(log^k(n)))$.

The reverse question of representing a Π^0_1 class of prescribed complexity by a decidable or axiomatizable theory is more interesting. We shall modify the classic proof of Ehrenfeucht [13] to give a converse of Theorem 2.3. As in the Theorem 2.3, there is no solid connection in the axiomatizable case between the complexity of the class and the complexity of the corresponding theory.

Theorem 2.4 *For any Π^0_1 class Q and for any time constructible function f which is nondecreasing and unbounded, there is a propositional theory Γ with a $DTIME(O(f))$ set of axioms such that Q represents the set of complete consistent extensions of Γ.*

Proof: We can establish the desired result by modifying the proof of the previous theorem. Given a computable tree S such that $Q = [S]$, first define a $DTIME(O(f))$ time tree T with $Q = [T]$ as follows. First put the empty sequence in T. Next, for any given $\sigma = (\sigma_1, \ldots, \sigma_n) \in \{0,1\}^n$, run the computations to test whether $(\sigma_1, \ldots, \sigma_i) \in S$ for $i = 1, 2, \ldots, n$ in order for $f(n)$ steps. If there is some i such that we have completed the computation to test whether $(\sigma_1, \ldots, \sigma_i) \in S$ and $(\sigma_1, \ldots, \sigma_i) \notin S$, then we declare that $\sigma \notin T$. Otherwise, we put $\sigma \in T$. Now it is easy to see that if $x = (x_1, x_2, \ldots)$ is an infinite path in $[S]$, then $(x_1, \ldots, x_n) \in T$ for all n so that $x \in [T]$. However, if $x \notin [S]$, then there is some m such that $(x_1, \ldots, x_m) \notin S$. Since f is unbounded, there is an n large enough so that we can compute the computations to test whether $(x_1, \ldots, x_i) \in S$ for $i = 1, \ldots, m$ with in $f(n)$ steps. It follows that if $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a string which extends $(x_1, \ldots, x_m)$, then $\sigma \notin T$ so that $x \notin [T]$. Thus T is a $DTIME(O(f))$ tree such that $[T] = [S]$ so that $Q = [T]$. Then Q represents the set of complete consistent extensions of the theory with a $DTIME(O(f))$ set Δ of axioms defined as in the proof of Theorem 2.3. $\Box$

The results for decidable trees are somewhat surprising. Let us first give a few definitions. Recall that SAT is the set of satisfiable, or consistent, propositional sentences and is the standard NP-complete set.

Theorem 2.5 *The following are equivalent:*

(i) $P = NP$;

(ii) *Every P-decidable tree represents the set of complete consistent extensions of some theory which is P-decidable in tally.*

Proof: [(ii) $\to$ (i)] Let $T = \{0,1\}^*$ and suppose that Δ is a theory which is P-decidable in tally such that $\{0,1\}^\omega = [T]$ represents the set of complete consistent extensions of Δ. Then it is easy to see that $SAT(\Delta) = SAT$. But this means that

$$\phi \in SAT \iff \neg[\Delta \vdash \neg\phi]. \tag{1}$$

Since Δ is P-decidable in tally, (1) would imply that SAT is polynomial time and hence $P = NP$.

[(i) $\to$ (ii)] Next suppose that $P = NP$ and let T be a P-decidable tree. Let $\phi(A_0, \ldots, A_n)$ be a propositional formula whose propositional letters are a a subset of $\{A_0, \ldots, A_n\}$ and which contains A_n. The canonical theory Δ such that $[T]$ represents the set of complete consistent extensions of Δ is defined by

$$\Delta \vdash \phi(A_0, \ldots, A_n) \iff (\forall \sigma \in T \cap \{0,1\}^n)(B(\sigma) \vdash \phi(A_0, \ldots, A_n)).$$

We will show that $SAT(\Delta)$ is NP and hence in P by our assumption. In tally, $n \leq |B(\sigma)| \leq 2n^2$, so that for each $\sigma \in \{0,1\}^n$, we can test whether $B(\sigma) \vdash \neg\phi(A_0, \ldots, A_n)$ in polynomial time in the length of $\phi(A_0, \ldots, A_n)$. Thus we can test $\phi(A_0, \ldots, A_n) \in SAT(\Delta)$ in the usual NP fashion, by guessing a string σ of length n and checking that $\sigma \in T$ and $B(\sigma)$ implies $\phi(A_0, \ldots, A_n)$. Thus $SAT(\Delta)$ is in P. $\Box$

The corresponding result does not follow relative to the binary representation of theories. That is, the direction [(ii) $\to$ (i)] still holds, since the SAT problem is NP-complete in either tally or binary. However, the argument given for the reverse direction only shows that $SAT(\Delta)$ is $DTIME(2^{O(1)})$-decidable in binary. This is due to the fact that a short formula ϕ with a high numbered variable, such as a propositional variable A_{2^n-1} requires us to check whether $B(\sigma) \vdash \phi(A_0, \ldots, A_n)$ for $|\sigma| = 2^n - 1$ which would require time of order 2^n

since $|B(\sigma)| \geq 2^n$. Thus since $|A_{2^n-1}| = n+1$, such a check would require exponential time in $|\phi|$.

Next we consider the problem of relating the complexity of a propositional theory to the complexity of particular complete consistent extensions.

Theorem 2.6 *Relative to the tally representation, the following hold.*

(i) *Any P-decidable theory has a P-decidable complete consistent extension.*

(ii) *Any $PSPACE$-decidable theory has a $PSPACE$-complete consistent extension.*

(iii) *Any $DEXT$-decidable theory has an $EXPTIME$-decidable complete consistent extension.*

(iv) *Any $EXPTIME$-decidable theory has an $EXPTIME$-decidable complete consistent extension.*

(v) *Any $EXPSPACE$-decidable theory has an $\bigcup_{k\in\omega} DSPACE(2^{n^k})$-decidable complete consistent extension.*

Proof: Let Δ be the given theory. A complete consistent extension Γ can be defined as follows.

Stage 0. Put $A_0 \in \Theta$ and let $B_0 = A_0$ if $\Delta \vdash A_0$ and otherwise put by $\neg A_0 \in \Theta$ and let $B_0 = \neg A_0$.

Stage $n+1$ Put $A_{n+1} \in \Theta$ if $\Delta \vdash B_n \rightarrow A_{n+1}$ and let $B_{n+1} = B_n \wedge A_{n+1}$ and otherwise put $\neg A_{n+1} \in \Theta$ and let $B_{n+1} = B_n \wedge \neg A_{n+1}$.

It is clear that the resulting set $\Gamma = Cn(\Theta)$ is a complete consistent extension of Δ. It remains to estimate the complexity of Γ. At stage $n+1$, we apply the algorithm for the decidability of Δ to the sentence $B_n \rightarrow A_{n+1}$, which has length of order n^2. We then have the following. (i) Suppose Δ is P-decidable, say it take time $c|\phi|^k$, for some c and k, to determine whether $\Delta \vdash \phi$. Then it takes time on the order of an^{2k}, for some constant a, to decide if $\Delta \vdash B_n \rightarrow A_{n+1}$ so that stage $n+1$ requires time on the order of bn^{2k}, for some constant b, to compute stage n. It thus takes time on the order of $\sum_{i=1}^{n+1} bi^{2k} \leq (bn^2k)^2$ to compute stages $1, \ldots, n+1$. Hence, for any $n \geq 0$, it requires time on the order of b^2n^{4k} to determine which of A_{n+1} or $\neg A_{n+1}$ is in Θ. Finally given any sentence ϕ, we can decide whether $\phi \in \Gamma$ within the required complexity because once we know for each A_i which occurs in ϕ which of A_i or $\neg A_i$ is in Γ, it only takes polynomial many steps to check if $\Gamma \vdash \phi$.

Items (ii) to (v) are straightforward modifications of (i). $\square$

We note that if Δ is LIN-decidable, our argument would only show that Δ had a $DTIME(cn^2)$-decidable complete consistent extension for some c.

Theorem 2.7 *Relative to the binary representation, the following hold.*

(i) *Any P-decidable theory has an $DEXT$-decidable complete consistent extension.*

(ii) *Any $PSPACE$-decidable theory has an $EXPSPACE$-decidable complete consistent extension.*

(iii) *Any $EXPTIME$-decidable theory has an $DOUBEXT$-decidable complete consistent extension.*

(iv) *Any $EXPSPACE$-decidable theory has an $DOUBEXPSPACE$-decidable complete consistent extension.*

Proof: Define the complete consistent extension Γ of the given consistent theory Δ as in the proof of Theorem 2.6. The difference relative to the binary representation the length of B_n is of order $n \cdot log\ n$ and also the number of steps n is exponential in the length $1 + bin(n)$ of A_n.

Thus, for example, if Δ is P-decidable, then stage $n+1$ still requires $p(n)$ steps for some polynomial p and there is a polynomial q such that we can decide if A_n is in Θ in $q(n)$. Thus if $q(n)$ is of the form cn^k for some constant c and integer k, then it requires $c(2^{log_2(n)})^k \leq 2^{K|A_n|}$ steps to decide if A_n is Γ for some constant K. It easily follows that Γ is $DEXT$-decidable. The arguments for parts (ii) to (iv) are straightforward modifications. $\square$

Next we show next that this difference in the complexity of the complete consistent extension between the tally and binary representations is necessary.

We need to consider an effective enumeration of the polynomial time functions on $\mathcal{L}$. Let ϕ_e be the e-th computable function and let $\phi_{e,s}(x)$ be the result, if any, of running the e-th Turing machine on input x for s-steps. Then we let $\pi_e(x) = \phi_{e,(1+|x|)^e}(x)$ if defined, and 0 if not. Clearly each function π_e is polynomial time. On the other hand, given e and r such that ϕ_e is a total function such that the e-th Turing machine always computes $\phi_e(x)$ in time $\leq (1+|x|)^r$, we can always choose $d > r$ such that for all x, $\phi_d(x) = \phi_e(x)$ and the two computations take exactly the same number of steps, so that $\phi_d(x)$ is computed in time $\leq (1+|x|)^r < (1+|x|)^d$. Thus $\phi_e = \pi_d$ and hence $\pi_0, \pi_1, \ldots$ is an effective list of all polynomial time functions.

Theorem 2.8 (i) *There is a propositional theory which is NP-decidable in binary but has no P-decidable complete consistent extension in binary.*

(ii) *There is a propositional theory which is $DEXT$-decidable in binary but has no $EXPTIME$-decidable complete consistent extension in binary.*

Proof: (i) We first define a P-decidable tree T such that $[T] \neq \emptyset$ but there is no path $\delta \in [T]$ which is computable in polynomial time. Given a string $\sigma \in \{0,1\}^k$, we describe the algorithm for testing whether $\sigma \in T$. For all e such that $2^{2^e} \leq k$, compute the string $\delta_e = (\pi_e(0), \ldots, \pi_e(2^{2^e} - 1))$. If there is an e with $2^{2^e} \leq k$ and $\delta_e \sqsubseteq \sigma$, then σ is not in T and if there is no such e, then σ

is in T. Let us compute the time required for this procedure. For each e, the computation of $\pi_e(0), \ldots, \pi_e(2^{2^e} - 1)$ requires time

$$|bin(0)|^e + |bin(1)|^e + \cdots + |bin(2^{2^e} - 1)|^e \leq 2^{2^e} \cdot (2^e)^e = 2^{2^e + e^2} \leq 2^{2 \cdot 2^e}.$$

If we let e be the largest such that $2^{2^e} \leq k$, then there are $e+1$ such computations to be done with a total computation time $\leq (e+1)2^{2 \cdot 2^e} \leq 2^{3 \cdot 2^e} \leq k^3$. The required comparisons with σ do not significantly increase the time so that it easily follows that T is a polynomial time tree. Observe that T has no dead ends by the following. Suppose $\sigma \in \{0,1\}^k \cap T$ so that $2^{2^e} \leq k \rightarrow \neg\delta_e \sqsubseteq \sigma$. Then $\sigma^\frown i \in T$ unless $k + 1 = 2^{2^{e+1}}$ and $\sigma^\frown i = \delta_{e+1}$, which can happen for at most one value of i.

Let Δ be the theory such that $[T]$ represents the set of complete consistent extensions of Δ. That is, as described above, $\phi(A_0, \ldots, A_{n-1}) \in SAT(\Delta)$ if and only if there is a path $\sigma \in T \cap \{0,1\}^n$ such that $B(\sigma) \vdash \phi$. We claim that $SAT(\Delta)$ is NP.

First we show that if ϕ is a conjunction of literals, then we can decide in polynomial time whether $\phi \in SAT(\Delta)$. That is, suppose that

$$C = B_1 \ \wedge \ B_2 \ \wedge \cdots \wedge B_k$$

is a conjunction of literals where each B_i is either A_{n_i} or $\neg A_{n_i}$ for some $n_1, \ldots, n_k$. Let $B_C = \{B_1, \ldots, B_k\}$ and $n = max(\{n_1, \ldots, n_k\}) + 1$. Suppose that $2^{2^f} \leq n < 2^{2^{f+1}}$. For any string γ, we will say that γ is consistent with C if and only if for all $n_i < |\gamma|$, $A_{n_i} \in B_C$ implies $\gamma(n_i) = 1$ and $\neg A_{n_i} \in B_C$ implies $\gamma(n_i) = 0$. Then it is clear that $C \in SAT(\Delta)$ if there is a node $\gamma \in T$ with length at least $2^{2^{f+1}} - 1$ such that γ is consistent with B_C. Note that $f \leq log_2(log_2(n))$ and $log_2(n) \leq |C|$ since either A_{n-1} or $\neg A_{n-1}$ occurs in C and hence

$$f \leq log_2(|C|). \tag{2}$$

Next let m be the least r such there exists exactly $f+1$ elements $s < r$ such that neither A_s nor $\neg A_s$ are elements of B_C. Note C has at least $m - f - 1$ literals so that $|C| > 2(m - f - 1) - 1 = 2m - 2f - 3$. Hence $m \leq (|C| + 2f + 3)/2 \leq (|C| + log_2|C| + 3)/2 \leq 2|C|$. It requires m scans of C to find the $f+1$ elements $s_1 < \ldots s_{f+1} < m$ such that neither A_{s_j} nor $\neg A_{s_j}$ occur in B_C. Since $m \leq 2|C|$, we can find m and $s_1, \ldots, s_{f+1}$ in polynomial time in $|C|$.

Let $[m] = \{0, \ldots, m-1\}$ and consider the 2^{f+1} strings γ of length m such that γ is consistent with C. We can easily compute all such γ in polynomial time in $|C|$ because $\gamma(i) = 1 \iff A_i \in B_C$ if $i \in [m] - \{s_1, \ldots, s_{f+1}\}$ and the values of $\gamma(s_j) \in \{0,1\}$ for $j = 1, \ldots, f+1$. There are $2^{f+1} = 2 \cdot 2^f \leq 2 \cdot 2^{log_2(|C|)} \leq 2|C|$ such strings and hence we can test all such strings for membership in T in polynomial time in $|C|$ since T is a polynomial time tree.

We claim that $C \in SAT(\Delta)$ if and only if there is a γ of length m which is consistent with C which is in T so that we can non-deterministically test whether $C \in SAT(\Delta)$ in polynomial time in $|C|$. Our claim follows from the following lemma. Let $\{0,1\}^{\leq n} = \{\sigma \in \{0,1\}^* : |\sigma| \leq n\}$. For any string

$\sigma \in \{0,1\}^{\leq n}$, let $Ext_n(\sigma) = \{\gamma \in \{0,1\}^{<n} : \sigma \sqsubseteq \gamma\}$. Note that for our tree T, $T_n = T \cap \{0,1\}^{\leq n} = \{0,1\}^{\leq n} - \bigcup_{e=0}^{f} Ext(\delta_e)$. Suppose that E and F are disjoint subsets of $[n]$. Then we say that $\gamma \in \{0,1\}^n$ separates the pair E, F if $\gamma(i) = 1$ if $i \in E$ and $\gamma(i) = 0$ if $i \in F$. Note that if $E = \{n_i : A_{n_i} \in B_C\}$ and $F = \{n_i : \neg A_{n_i} \in B_C\}$, then $\gamma \in \{0,1\}^n$ separates the pair E, F if and only if γ is consistent with C. For each $j < n$, we let $E_j = E \cap [j]$ and $F_j = F \cap [j]$. Observe that if γ separates (E, F), then, for all $j \leq |\gamma|$, $\gamma\lceil j$ separates (E_j, F_j). With this notation, it is clear that our next lemma will show that $C \in SAT(\Delta)$ if and only if there is a $\gamma \in T$ of length m which is consistent with C, and hence that we can decide if $C \in SAT(\Delta)$ in polynomial time in $|C|$.

Lemma 2.9 *Suppose that $T = \{0,1\}^{\leq n} - \bigcup_{i=0}^{f} Ext_n(\beta_i)$ where $\beta_0, \dots, \beta_f$ are elements of $\{0,1\}^{\leq n}$ and E and F are disjoints subsets of $[n]$ such that $|E| + |F| \leq n - f - 1$. Let $j \leq n$ be such that $|[j] - (E_j \cup F_j)| \geq f + 1$ and suppose that there is $\delta \in \{0,1\}^j \cap T$ such that δ separates the pair (E_j, F_j), then there is a $\gamma \in T$ such that γ separates the pair (E, F)*

Proof: For any set $D \subseteq [n]$, we let $\gamma_{D,n} \in \{0,1\}^n$ be the string of length n such that $\gamma_d(i) = 1 \iff i \in D$. Suppose $f = 0$ and let a be the least element of $[j] - (E_j \cup F_j)$. There are two cases.

Case 1.a $|\beta_0| \leq a$.
In this case, it cannot be the case that β_0 separates the pair $(E_{|\beta_0|}, F_{|\beta_0|})$ since otherwise by our choice of a, $E_{|\beta_0|} \cup F_{|\beta_0|} = [|\beta_0|]$ and hence every sequence of length j separating (E_j, F_j) would extend β_0 and therefore could not be in T. Thus the string $\gamma_{E,n}$ is in T and separates the pair (E, F).

Case 1.b $|\beta_0| > a$.
In this case, if $\beta_0(a) = 1$, then the string $\gamma_{E,n}$ is in T since $\gamma_{E,n}(a) = 0$ and $\gamma_{E,n}$ separates the pair (E, F). Otherwise, $\beta_0(a) = 0$ and the string $\gamma_{E \cup \{a\},n}$ is in T and separates the pair (E, F).

Now assume by induction that the lemma is true for $f \leq e$ and consider the case where $f = e+1$. Assume that we have numbered the strings $\beta_0, \dots, \beta_{e+1}$ so that $|\beta_0| \leq |\beta_1| \leq \cdots \leq |\beta_{e+1}|$. Again let a be the least element of $[j] - (E_j \cup F_j)$. There are three cases.

Case 2.a $|\beta_0| \leq a$.
Again it cannot be the case that β_0 separates the pair $(E_{|\beta_0|}, F_{|\beta_0}|)$. Thus any string γ which separates the pair (E, F) is automatically in $\{0,1\}^{<n} - Ext_n(\beta_0)$. Thus we can apply the induction hypothesis to the tree $T' = \{0,1\}^{\leq n} - \bigcup_{i=1}^{e+1} Ext_n(\beta_i)$ to conclude that there is a $\gamma \in T'$ such that γ separates the pair (E, F). But then we know that $\gamma \notin Ext_n(\beta_0)$ so that in fact γ is in T.

Case 2.b $|\beta_0| > a$ and there is a $j \in \{0,1\}$ such that $\beta_i(a) = j$ for $i = 0, \dots, f$.
In this case, if $j = 1$, then the string $\gamma_{E,n}$ is in T since $\gamma_E(a) = 0$ and $\gamma_{E,n}$ separates the pair (E, F). Otherwise, $j = 0$ and the string $\gamma_{E \cup \{a\},n}$ is in T and separates the pair (E, F).

Case 2.c Not case 2.a or case 2.b.
Thus $|\beta_0| > a$ and there exist strings β_u and β_v with $u, v \leq e+1$ such that $\beta_u(a) = 1$ and $\beta_v(a) = 0$. Now consider the string δ of the length j in T such that δ separates the pair (E_j, F_j). If $\delta(j) = 1$, then, by induction, the lemma holds for the pair $(E \cup \{a\}, F)$ relative to the tree $T'' = \{0,1\}^{\leq n} - \bigcup_{i \in [e+1]-\{r:\beta_r(a)=0\}} Ext_n(\beta_i)$. That is, $[e+1] - \{r : \beta_r(a) = 0\}$ is a set of size at most e and $[j] - (E \cup \{a\} \cup F)$ is of size at least e+1 and hence we can apply the lemma by induction. Hence there is a string γ of length n in T' which separates the pair (E, F) such that $\gamma(a) = 1$. But then $\gamma \notin \bigcup_{i \in \{r:\beta_r(a)=0\}} Ext_n(\beta_i)$ and hence γ must be in T. The case when $\delta(j) = 0$ is similar. This completes the proof of the lemma. $\square$

The non-deterministic polynomial time procedure for checking whether an arbitrary sentence $\phi(A_{n_1}, \ldots, A_{n_k})$ is in $SAT(\Delta)$ is the following. Guess a conjunction $C = B_1 \wedge B_2 \wedge \cdots \wedge B_k$ of literals where B_i is either A_{n_i} or $\neg A_{n_i}$ for $i = 1, 2, \ldots, k$. (This is equivalent to guessing a sequence in $\{0,1\}^k$ where $k < |\phi|$.) The verification procedure has two parts. First, verify that $C \vdash \phi$, which is accomplished simply by substituting the values for A_j indicated by C. Second, verify that $C \in SAT(\Delta)$ as indicated above. Thus $SAT(\Delta)$ is in NP and hence Δ is NP-decidable.

However, the only complete consistent extensions of Δ are of the form $Cn(\{A_i : \pi(i) = 1\} \cup \{\neg A_i : \pi(i) = 0\})$ where $\pi = (\pi(0), \pi(1), \ldots)$ is an infinite path. But it is easy to see that T has no infinite path which is in P since we ensured that for every e, there is no path extending $\pi_e(0), \ldots, \pi_e(2^{2^e} - 1))$. Now T has infinite paths since one can construct a path $\pi = (\pi(0), \pi(1), \ldots)$ by a simple diagonalization argument such that $\pi_e \not\sqsubseteq \pi$ for any e. Thus Δ has complete consistent extensions but it does not have any complete consistent extensions which are in P.

(ii) The proof follows the outline of part (i), so we just sketch the differences.

First we have to define the e-th $EXPTIME$ function, π'_e, as follows. Let $f(e,t) = 2^{(1+t)^e}$ and let $\pi'_e(x) = \phi_{e,f(e,|x|)}(x)$ if defined, and 0 if not.

The tree T is defined so that for $\sigma \in \{0,1\}^k$, we compute $\pi_e(i)$ for all e with $2^{e^2} < k$ and all i with $i < 2^{e^2}$ and check whether $\pi_e(i) = \sigma(i)$ for all $i < 2^{e^2}$. Then we put $\sigma \in T$ if and only if there is no e such that σ and π_e agree up to 2^{e^2}.

Next we compute the maximum time required to check $\sigma \in T$. For the computation of $\pi'_e(i)$, we have $|bin(i)| \leq e^2 \leq 2^e - 1$, so that the time required is $\leq 2^{(2^e)^e} = 2^{2^{e^2}} \leq 2^k$. Since we have $i < 2^{e^2} < k$ and $e < k$, there are fewer than k^2 such computations. This gives a total time on the order of 2^{2k}, so that T is exponential time, as desired.

The exponential time theory Δ is again chosen so that $[T]$ represents the set of complete consistent extensions of Δ. To test whether a conjunction $C = B_1 \wedge B_2 \wedge \cdots \wedge B_k$ is in $SAT(\Delta)$, we check for all e such that $2^{e^2} \leq k$, whether A_i occurs in C for every $i < 2^{2^e}$ and whether the corresponding string τ_e is in T, as described in part (i).

The exponential time procedure for checking whether $\phi(A_{n_1}, \ldots, A_{n_k}) \in SAT(\Delta)$ again consists of guessing truth values for each A_{n_i} and then verifying that the corresponding conjunction C is in $SAT(\Delta)$ and implies ϕ. Now there are only 2^k different choices for the conjunction C and each verification can be done in exponential time of order 2^{2k}. $\square$

3 Complexity Theoretic Completeness Theorem for Predicate Logic

The Completeness Theorem of Gödel showed that every consistent first order theory has a complete consistent extension and has a model. Henkin's construction is effective and hence the Computable Completeness Theorem holds, that is, every decidable consistent first order theory has a complete consistent decidable extension and has a computable model. A theory which is just computably axiomatizable does not necessarily have a computable complete consistent extension. For example, Peano Arithmetic is computably axiomatizable but does not have a computable complete consistent extension. In the last section, we saw that for propositional theories, a polynomial time decidable theory in tally always has a polynomial time complete consistent extension and that an NP time theory in binary does not always have a polynomial time complete consistent extension. In this section, we consider the same questions for first order logical theories.

We now give the main theorem of the present paper which is concerned with the problem of starting with a decidable first order theory T whose decision problem lies in some well known complexity class such as P, $PSPACE$, etc. and constructing a decidable complete consistent extension $\mathcal{A}$ of T whose decision problem lies in a slightly higher complexity class along with a decidable model $\mathcal{M}$ of $\mathcal{A}$. Since any consistent, computable relational theory has a computable model and therefore has a linear time computable model by [3], one expects to obtain a model $\mathcal{M}$ which lies in a lower complexity class than the complexity class of $\mathcal{A}$.

There are three components to our complexity theoretic version of the Henkin construction. The central construction will show how a decidable first-order theory T in a given complexity class may be extended to a complete consistent theory $\mathcal{A}$ of slightly higher complexity class. For example, a $PSPACE$ decidable theory can be expanded to an $EXPSPACE$-decidable complete theory. An infinite set of new constants $c_0, c_1, \ldots$ are introduced and we will use these constants to help construct the $EXPSPACE$ model M of our complete theory $\mathcal{A}$.

Next we study the question of whether the $EXPSPACE$-decidable model M is isomorphic to a model which is in a lower complexity class such as $PSPACE$. This seems reasonable given the result described above that every decidable relational theory has a polynomial time model. In our case, the standard Henkin construction of our $EXPSPACE$-decidable complete consistent theory $\mathcal{A}$ is modified by reordering the sentences so that for any fixed relation, say $R(x)$, the sentences of the form $R(c_n)$ occur in a polynomial sequence, say as the kn^2-th sentence in our modified list. This fact will allow us to show that complexity of the model M is in $PSPACE$.

The final part of our construction deals with the problem of ensuring that M has a standard universe. We know that there are computable relational models which have no primitive recursive computable models with a standard universe so it is somewhat surprising that here we can obtain a model M of $\mathcal{A}$ with a standard universe. Here the key idea is to assume that our theory has an infinite model. Thus the sentences of the form $c_i \neq c_j$, which make all of the constants distinct, are consistent with the given theory and we can simply add all such sentences to the theory in the beginning of the construction. Note that if the theory has a finite model, then of course there is a model in P.

Theorem 3.1 **(i)** *Suppose Δ is a $PSPACE$-decidable theory in binary (tally). Then Δ has a $PSPACE$ computable model with universe $Bin(\omega)$ ($Tal(\omega)$) which is $EXPSPACE$-decidable in binary (tally).*

(ii) *Suppose Δ is a P-decidable theory in binary (tally). Then Δ has an P-computable model with universe $Bin(\omega)$ ($Tal(\omega)$) which is a $DEXT$-decidable in binary (tally).*

(iii) *Suppose Δ is an $EXPSPACE$-decidable theory in binary (tally). Then Δ has an $\bigcup_{k>0} DSPACE(2^{n^k})$-computable model with universe $Bin(\omega)$ ($Tal(\omega)$) which is $DOUBEXPSPACE$-decidable in binary (tally).*

(iv) *Suppose Δ is an $DEXT$-decidable theory in binary (tally). Then Δ has an $EXPTIME$-computable model with universe $Bin(\omega)$ ($Tal(\omega)$) which is $DOUBEXT$-decidable in binary (tally).*

Proof: The proof uses a Henkin-style construction of the model. We will give the proof of part (i) and then indicate the necessary changes for the other parts.

Suppose that we are given a $PSPACE$-decidable first order theory Δ. Assume that the underlying language $\mathcal{L}$ is finite or countably infinite. We assume the variables of the underlying language are $x_0, x_1, \ldots$. We let $\mathcal{L}^*$ consist of $\mathcal{L}$ plus countable infinitely many new constant symbols $c_0, c_1, \ldots$. In the binary (tally) representation of $\mathcal{L}$ or $\mathcal{L}^*$, the n-th variable x_n is written as $x^\frown bin(n)$ ($x^\frown tal(n)$), the n-th constant symbol a_n is written as $a^\frown bin(n)$ ($a^\frown tal(n)$), the n-th relation symbol R_n is written as $R^\frown bin(n)$ ($R^\frown tal(n)$), and the n-th function symbol f_n is written as $f^\frown bin(n)$ ($x^\frown tal(n)$). In either the binary or tally representation of $\mathcal{L}^*$, the standard algorithm will determine whether a sequence of symbols ϕ in our language is a well-formed formula or is a sentence in time on the order of $O(|\phi| \cdot log(|\phi|)$. We order the underlying symbols of $\mathcal{L}^*$ in increasing order as $(,), \neg, \vee, \wedge, \rightarrow, \leftrightarrow, \forall, \exists, =, R, f, a, c, 0, 1$.

We will construct a complete consistent extension of Δ in the expanded language and then define the $EXPSPACE$-model using the set of constants as the domain. The proof will be established in a series of claims. The first task is to enumerate the formulas of the expanded language.

Let $\phi_0, \phi_1, \ldots$ enumerate the sentences of $\mathcal{L}^*$ first by length and then lexicographically within sentences of the same length. We first establish that there are on the order of 2^{cn} sentences of each length n.

Claim 3.2 *For some constant a and integer N, there are $\geq 2^{a(n+1)}$ sentences of length n for all $n \geq N$.*

Proof: Suppose that there is a sentence ϕ of length k. Then the sentences $(\forall x_i)(\phi)$ and $(\exists x_i)(\phi)$, for $i \in \omega$, will show that there are at least 2 sentences of length n for all $n \geq l = k + 7$. Now for any $m \geq l$, the sentences $(\exists x_0)(\phi)$ and $(\forall x_0)(\phi)$ will show that if there are p sentences of length m, there are at least $2p$ sentences of length $m + 7$. Hence for every $k \geq l$, there are at least $2^k p$ sentences of length $l + 7k$. Thus for any $n \geq l$, there are at least $2^{\frac{n-l}{8}+1}$ sentences of length n. $\Box$

We can deduce from Claim 3.2 an upper bound on the length of the i-th sentence ϕ_i. For any i, let $log(i)$ be the least k such that $2^k \geq i$, so that $i \leq 2^{log\ i} < 2i$.

Claim 3.3 *There is a constant b such that for all i, $|\phi_i| \leq b \cdot log(i)$.*

Proof: Choose a and N by Claim 3.2 so that for $n > N$, there are $\geq 2^{an}$ sentences of length $n - 1$. Let $|\phi_i| = n$ where $n > N$. Then $i > 2^{an}$, so that $log(i) > an$ and $n < \frac{1}{a}\ log(i)$. Now let b be the maximum of $\frac{1}{a}$ and the number of sentences of length $\leq N$. $\Box$

Claim 3.4 *For some constant c, there are $\leq 2^{cn}$ formulas of length $\leq n$, for sufficiently large n.*

Proof: This follows from having a finite alphabet. Thus if there are k symbols in the alphabet, then clearly there are k^n strings of length n and thus $\leq k^{n+1}$ strings of length $\leq n$. Taking c so that $2^c \geq k$, we have the claim. $\Box$

Claim 3.5 *For some constant d, there is a procedure which computes the list of sentences $\phi_0, \ldots, \phi_n$ in time $\leq (2n)^d$.*

Proof: By Claim 3.3, we know that $|\phi_n| \leq b \cdot log\ n$. Now the number of strings of length $b \cdot log\ n$ is $\leq 2^{b \cdot log\ n} < (2n)^b$ so that the total number of strings of length $\leq b \cdot log\ n$ is $< (2n)^{b+1}$. Thus to find ϕ_n, we enumerate all strings of length $\leq b \cdot log\ n$, test whether each one is a sentence and keep a list of the sentences until we reach the n-th one. Since each string being tested has length $\leq b \cdot log\ n$, each test can be done in time of order $\leq log\ n \cdot log\ log\ n$, so that the total time required is of order $\leq (2n)^{b+1} \cdot log\ n \cdot log\ log\ n \leq (2n)^{b+3}$. $\Box$

Observe that none of the claims 3.3-3.6 depends on whether we write our formulas in tally or in binary notation.

We now sketch the Henkin-style construction needed to obtain the complete consistent extension and model of Δ. The construction will proceed in stages $s \geq$

0. The standard Henkin construction is modified so that all of the new constants are kept distinct. That is, we can assume that our theory has only infinite models since otherwise Δ has a model in P whose theory is $NSPACE(O(n))$-decidable so that Δ would have a $DEXT$-decidable complete consistent extension. So assume that Δ has only infinite models. Thus $\Delta^+ = \Delta \cup \{c_i \neq c_j : i < j\}$ is consistent. We will ensure that Δ^+ is included in our complete consistent extension.

Claim 3.6 *1. If Δ is $PSPACE$-decidable, then Δ^+ is $PSPACE$-decidable.*

2. If Δ is P-decidable, then Δ^+ is P-decidable.

3. If Δ is $EXPSPACE$-decidable, then Δ^+ is $EXPSPACE$-decidable.

4. If Δ is $DEXT$-decidable, then Δ^+ is $DEXT$-decidable.

Proof: Let $|\phi_e| = n$ and suppose that $c_{i_1}, \dots, c_{i_k}$ is a list of the new constants occurring in ϕ_e. Then $k < n$ and, for all $s < t < k$, $|c_{i_s} \neq c_{i_t}| \leq n+3$. It follows that the conjunction ψ_e of all such inequalities has length $< (n+3)^3$ and that we can construct ψ_e from ϕ_e in polynomial time. Parts (1)-(4) now easily follow from the fact that

$$\Delta^+ \vdash \phi_e \iff \Delta \vdash (\psi_e \to \phi_e).$$

□

The complete theory $\Gamma = \{\delta_e : e = 0, 1, \dots\}$ is defined in stages as follows.

Stage $s = 2e$: See whether $\Delta^+ \vdash (\delta_0 \wedge \cdots \wedge \delta_{2e-1}) \to \phi_e$. If so, then $\delta_s = \phi_e$. If not, then $\delta_s = \neg\phi_e$. (For $e = 0$, just check $\Delta^+ \vdash \phi_0$.)

Stage $s = 2e + 1 > 1$: See whether δ_e has the form $(\exists x_n)\theta(x_n)$. If not, then just let $\delta_s = \delta_{s-1}$. If so, then we will select one of the new constants c_i and let $\delta_s = \theta(c_i)$. We cannot simply choose the first constant which has not been used in any of $\delta_0, \dots, \delta_{s-1}$, because it may be that we have already declared that $\theta(c_0) \in \Gamma$ and Δ ensures that a unique element satisfies $\theta(x)$. Thus we first check whether $\theta(c_j)$ for any of the constants c_j occurring in $\delta_0, \dots, \delta_{s-1}$ and let $c_i = c_j$ for the least such j if there is one. Otherwise, we let c_i be the first constant which has not been used in any of $\delta_0, \dots, \delta_{s-1}$.

It follows as usual that Γ is a complete consistent extension of Δ^+. Our next goal is to show that since Δ is $PSPACE$-decidable, Γ is $EXPSPACE$-decidable. We give a series of calculations.

Claim 3.7 *There is a constant r such that, for each s and for each c_t occurring in ϕ_s or in δ_s, $|c_t| \leq r \cdot log\ s$ in the binary case and $|c_t| \leq s^r$ in the tally case.*

Proof: Let b be given by Claim 3.3 so that $|\phi_s| \leq b \cdot log\ s$. Then for c_t in ϕ_s, we also have $|c_t| \leq b \cdot log\ s$ hence $log\ t \leq b \cdot log\ s$ and $t \leq s^b$. Now the total number of new constants added by stage s is fewer than s by the construction. Hence if $c_t \in \delta_s$, then $t < s^b + s < s^{b+1}$. Thus $|c_t| \leq s^{b+1}$ in tally and $|c_t| \leq (b+1) \cdot log(s)$ in binary. □

Claim 3.8 *There is a constant q such that for each s, $|\delta_s| \leq q \cdot log^2(s)$ in the binary case and $|\delta_s| \leq s^q$ in the tally case.*

Proof: Let $q = (b+1)(r+1) \geq b+r$, where b and r are the constants given by Claims 3.3 and 3.7.

We may omit the case where $\delta_s = \delta_{s-1}$, which leaves three cases:

Case 1: $s = 2e$ and $\delta_s = \phi_e$. Then $|\delta_s| = |\phi_e| \leq b \cdot log(e) \leq q \cdot log\ s$ by Claim 3.3.

Case 2: $s = 2e$ and $\delta_s = (\neg\phi_e)$. Then $|\delta_s| = |\phi_e| + 3 \leq 4b \cdot log(e) \leq 4q \cdot log(s)$ by Claim 3.3.

Case 3: $s = 2e+1$ and $\delta_s = \theta(c_t)$ where $\delta_e = (\exists x_i)(\theta)$.

In Case 3, consider the sentence δ_e. It is not a negation, so either $e = 2d$ and $\delta_e = \phi_d$ or $e = 2d+1$ and either $\delta_e = \delta_{e-1}$ or $\delta_e = \theta'(c_p)$, where $e = 2f+1$ and $\delta_f = (\exists x_h)\theta'$. It follows by induction that for some $d \leq e$, some $k \geq 1$ and some variables $x_{i_1}, \ldots, x_{i_k}$ and constants $c_{j_1}, \ldots, c_{j_k}$, $\phi_d = (\exists x_{i_1})(\exists x_{i_2}) \ldots (\exists x_{i_k})\theta(x_{i_1}, \ldots, x_{i_k})$ and $\delta_s = \theta(c_{j_1}, \ldots, c_{j_k})$.

By Claim 3.3, $|\phi_d| \leq b \cdot log(s)$.

First consider the tally case.

For each t, $|c_{j_t}| \leq s^r$ by Claim 3.7; let c_j have maximal length. Now there can be no more than $|\phi_d|$ occurrences of the constants c_{j_t} in δ_s, so that in the tally case,

$$|\delta_s| \leq |\phi_d||c_j| \leq b \cdot log(s)s^r \leq s^b s^r \leq s^q.$$

In the binary case, we can conclude that $|c_j| \leq r \cdot log(s)$, so that

$$|\delta_s| \leq |\phi_d||c_j| \leq b \cdot log(s) \cdot r \cdot log(s) \leq q \cdot log^2(s).$$

□

The key to the argument lies in the following bound on the complexity of the construction.

Claim 3.9 *Stage s of the construction takes space $O(s^k)$ for some fixed k.*

Proof: By Claim 3.5, we may assume that we have a list of the first s sentences $\phi_0, \ldots, \phi_{s-1}$. Then we first see which of the two cases s falls into, that is, we test whether s is even or not. This takes time $O(s)$. The construction of ϕ_s now falls into the two cases.

Case 1: $s = 2e$, where we use our *PSPACE* decidability algorithm to test $\Delta^+ \vdash (\delta_0 \wedge \cdots \wedge \delta_{s-1}) \rightarrow \phi_e$. By Claims 3.3 and 3.8,

$$|(\delta_0 \wedge \cdots \wedge \delta_{s-1}) \rightarrow \phi_e| \leq s \cdot s^q + b \cdot log(e) \leq s^{b+q+2}$$

in either binary or tally. If our algorithm runs in space $|\phi|^w$, then we need space at most $s^{w(b+q+2)}$ to decide whether to put ϕ_e or $\neg\phi_e$ into Γ at this stage in either the binary or the tally representation.

Case 2: For $s = 2e + 1$, $e > 1$: We read δ_e and determine whether it has the form $(\exists x_i)\theta(x_i)$. This again takes time at most $O(s^q)$. Next we test

$$\Delta^+ \vdash ((\delta_0 \wedge \cdots \wedge \delta_{s-1}) \rightarrow \theta(c_i))$$

for all c_i occurring in $\delta_0, \ldots, \delta_{s-1}$.

Then $|\theta(c_i)| \leq s^r s^q$ by Claims 3.7 and 3.8 and it follows as in Case 2 that each such test can be done in space $\leq (s \cdot s^q + s^{r+q})^w \leq s^{2w(r+q+2)}$. Finally, if necessary, we take the next c_t from our list and write $\delta_s = \theta(c_t)$, which takes additional time at most $\leq s^q(s^q + 1)$. $\square$

It follows that the total space required to complete the computations for stages 1 through s is of order s^k for some k.

Claim 3.10 *We can decide whether $\phi_e \in \Gamma$ using at most $2^{c|\phi_e|}$ space for some fixed c.*

Proof: Let $n = |\phi_e|$. By the construction, ϕ_e is decided by stage $s = 2e$. By Claim 3.4, we have $e \leq 2^{cn}$, Thus by Claim 3.9, the space required to decide ϕ_e is

$$\leq (2^{cn+1})^k = 2^{k(cn+1)}.$$

$\square$

By the usual Henkin argument, Γ is an $EXPSPACE$-decidable complete consistent extension of Δ^+ so that we can obtain a $EXPSPACE$-decidable complete consistent extension of Δ by restricting Γ to the language $\mathcal{L}$. Now we will describe the $EXPSPACE$ model. The universe of the tally model $\mathcal{A}$ is $Tal(\omega)$ and the universe of the binary model $\mathcal{B}$ is $Bin(\omega)$. The relations and functions of the structures $\mathcal{A}$ and $\mathcal{B}$ are defined using the theory Γ. That is, for any relation symbol R_m and any $tal(n_1), \ldots, tal(n_k) \in A$, we have

$$R_m^A(tal(n_1), \ldots, tal(n_k)) \iff \Gamma \vdash R_m(c_{n_1}, \ldots, c_{n_k}).$$

Then

$$\begin{aligned} |R_m(c_{n_1}, \ldots, c_{n_k})| &= |tal(n_1)| + \cdots + |tal(n_k)| + 2k + 2 + m + 1 \\ &\leq 5(|tal(n_1)| + \cdots + |tal(n_k)|) + m + 1. \end{aligned}$$

Since Γ is $EXPSPACE$-decidable and $|c_n| = n + 1 = |tal(n)| + 1$, it follows that R_m^A is $EXPSPACE$ computable.

For any function symbol F_p and any $tal(n_1), \ldots, tal(n_k), tal(n)$, we have

$$F_p^A(tal(n_1), \ldots, tal(n_k)) = tal(n) \iff \Gamma \vdash F_p(c_{n_1}, \ldots, c_{n_k}) = c_n.$$

Now let $m = |n_1 + \cdots + n_k|$ and consider the formula

$$\phi_{e(n_1,\ldots,n_k)} : (\exists x_1)(x_1 = F_p(c_{n_1}, \ldots, c_{n_k})).$$

If $e = e(n_1, \ldots, n_k)$, then $|\phi_e| < m+d$ for a fixed constant d so that by Claim 3.4, $e \leq 2^{c(m+d)}$. It follows from the construction that by stage $s = 2e \leq 2^{c(m+d)+1}$ we have $\delta_s = \phi_e$ and that by stage $2s+1 \leq 2^{c(m+d)+3}$, we have chosen a constant c_t and put $c_t = F_r(c_{n_1}, \ldots, c_{n_k})$ in Γ. It follows from Claim 3.9 that we can compute $c_t = F(c_{n_1}, \ldots, c_{n_k})$ in space $\leq (2s+1)^k \leq 2^{kc(m+d+4)}$ which makes F_p an $EXPSPACE$ computable function.

It then easily follows that the atomic diagram of $\mathcal{A}$ is in $EXPSPACE$.

For the binary structure $\mathcal{B}$, the relations R_m^B and functions F_p^B are similarly defined. The difference is that now $|c_n| = |bin(n)| + 1 = log(n) + 1$. The argument for the complexity of the relation R_m^B goes through as above. For the computation of the function $F_p^B(bin(n_1), \ldots, bin(n_k))$, let $m = |bin(n_1)| + \cdots + |bin(n_k)|$. If $e = e(n_1, \ldots, n_k)$, then $|\phi_e| < m + d$ for a fixed constant d so that by Claim 3.4, $e \leq 2^{c(m+d)}$. It follows from the construction that by stage $s = 2e \leq 2^{c(m+d)+1}$ we have $\delta_s = \phi_e$ and that by stage $2s+1 \leq 2^{c(m+d)+3}$, we have chosen a constant c_t and put $c_t = F(c_{n_1}, \ldots, c_{n_k})$ in Γ. It follows from Claim 3.9 that we can compute c_t in space at most $(2s+2)^k \leq 2^{kc(m+d+4)}$ which makes F_p an $EXPSPACE$ computable function.

Furthermore, it follows from Claim 3.7 that

$$|c_t| \leq 1 + r \cdot log(2s+1) \leq 2r \cdot log(2^{c(m+d)+3}) = 2rc(m+d) + 6r.$$

It then easily follows that atomic diagram of $\mathcal{B}$ is in $EXPSPACE$.

Next we deal with the problem of reducing the complexity of the model constructed above. We will describe the necessary modification to the proof of the main theorem given above to produce a polynomial space structure in case (i). First suppose that Δ is $PSPACE$-decidable in tally. The idea is to simply define a new enumeration of the formulas ϕ_e as follows. Let R_i be an enumeration of the relation symbols and let F_i be an enumeration of the function symbols. Fix some polynomial time pairing function $[,]_t : Tal(\omega) \times Tal(\omega) \to Tal(\omega)$ and extend $[,]_t$ to a n-tuples for $n \geq 3$ by the usual inductive definition $[tal(a_1), tal(a_2), \ldots, tal(a_n)]_t = [tal(a)_1, [tal(a_2), \ldots tal(a_n)]_t]_t$. Then, we let $\phi^*_{2e+1} = \phi_e$ and we let $\phi^*_{2[tal(0),tal(e),tal(k),tal(n_1),\ldots,tal(n_k)]_t}$ be $R_e(c_{n_1}, \ldots, c_{n_k})$ and we let $\phi^*_{2[tal(1),tal(e),tal(k),tal(n_1),\ldots,tal(n_k)]_t}$ be $(\exists x_0)(x_0 = F_e(c_{n_1}, \ldots, c_{n_k})$. If $2k$ is not of the form $[tal(0), tal(x)]_t$ or $[tal(1), tal(x)]_t$, then we let $\phi^*_{2k} = \phi_0$. If $2k$ is of the form $[tal(0), tal(e), tal(k), tal(x)]_t$ where R_e is either not in the language $\mathcal{L}$ or is R_e in $\mathcal{L}$ but is not a k-ary relation, then we let $\phi^*_{2k} = \phi_0$. Similarly, if $2k$ is of the form $[tal(1), tal(e), tal(k), tal(x)]_t$ where F_e is either not in the language $\mathcal{L}$ or is F_e in $\mathcal{L}$ but is not a k-ary function symbol, then we let $\phi^*_{2k} = \phi_0$.

Now relative to the tally representation, it is the case that for any fixed e, $R_e(c_{n_1}, \ldots, c_{n_k})$ is decided by stage $2[tal(0), tal(e), tal(k), tal(n_1), \ldots, tal(n_k)]_t$ and $F_e(c_{n_1}, \ldots, c_{n_k})$ is computed by stage

$$2(2[tal(0), tal(e), tal(k), tal(n_1), \ldots, tal(n_k)]_t) + 1$$

so that, by Claim 3.10, both the relations and the functions of $\mathcal{A}$ are polynomial space computable.

If Δ is $PSPACE$ decidable in binary, then we can use the same idea except that we use polynomial time pairing function $[,]_b : Bin(\omega) \times Bin(\omega) \to Bin(\omega)$. Then relative to the binary representation, it is the case that for any fixed e, $R_e(c_{n_1}, \ldots, c_{n_k})$ is decided by stage $2[bin(0), bin(e), bin(k), bin(n_1), \ldots, bin(n_k)]_b$ and $F_e(c_{n_1}, \ldots, c_{n_k})$ is computed by stage

$$4(2[bin(0), bin(e), bin(k), bin(n_1), \ldots, bin(n_k)]_b)^2 + 1,$$

so that, by Claim 3.10, both the relations and the functions of $\mathcal{B}$ are polynomial space computable.

This completes the proof of part (i).

For the other parts of the theorem, we need only re-examine the proof in Claim 3.9 and Claim 3.10. That is, we have the following.

Claim 3.11 *1. If Δ^+ is P-decidable, then stage s of the construction takes time $O(s^k)$ for some fixed k.*

2. If Δ^+ is $EXPSPACE$-decidable, then stage s of the construction takes time $O(2^{s^k})$ for some fixed k.

3. If Δ^+ is $DEXT$-decidable, then stage s of the construction takes time $O(2^{s^k})$ for some fixed k.

Proof: By Claim 3.5, we may assume that we have a list of the first s sentences $\phi_0, \ldots, \phi_{s-1}$. Then we first see which of the two cases s falls into, that is, we test whether s is even or whether $s = 2e + 1$ for some $e > 1$. This takes time $O(s)$. The construction of ϕ_s now falls into the two cases.

Case 1: $s = 2e$, where we use our decidability algorithm to test $\Delta^+ \vdash (\delta_0 \wedge \cdots \wedge \delta_{s-1}) \to \phi_e$. By Claims 3.3 and 3.8,

$$|(\delta_0 \wedge \cdots \wedge \delta_{s-1}) \to \phi_e| \leq s \cdot s^q + b \cdot log(e) \leq s^{b+q+2}.$$

If our algorithm runs in time $|\phi|^w$, then we need time at most $s^{w(b+q+2)}$ to decide whether to put ϕ_e or $\neg\phi_e$ into Γ at this stage in either the binary or the tally representation.

If our algorithm runs in space $2^{w|\phi|}$, then we need space at most $2^{ws^{(b+q+2)}}$ to decide whether to put ϕ_e or $\neg\phi_e$ into Γ at this stage in either the binary or the tally representation.

If our algorithm runs in time $2^{w|\phi|}$, then we need time at most $2^{ws^{(b+q+2)}}$ to decide whether to put ϕ_e or $\neg\phi_e$ into Γ at this stage in either the binary or the tally representation.

Case 2: For $s = 2e + 1$, $e > 1$: We read δ_e and determine whether it has the form $(\exists x_i)\theta(x_i)$. This again takes time at most $O(s^q)$. Next we test

$$\Delta^+ \vdash ((\delta_0 \wedge \cdots \wedge \delta_{s-1}) \to \theta(c_i))$$

for all c_i occurring in $\delta_0, \ldots, \delta_{s-1}$. In the tally representation, each $|c_i| \leq s^r$ by Claim 3.7. and thus there are at most s^r such tests for the tally representation. It follows that in either the tally or binary representation, $|\theta(c_i)| \leq s^r s^q$

If our algorithm runs in time $|\phi|^w$, then we need time at most $s^r(ss^q + s^r s^q)^w \leq s^{2w(r+q+2)}$ to decide whether to put ϕ_e or $\neg\phi_e$ into Γ at this stage in either the binary or the tally representation.

If our algorithm runs in space $2^{w|\phi|}$, then we need space at most $s^{2r}2^{w(ss^q+s^r s^q)}$ to decide whether to put ϕ_e or $\neg\phi_e$ into Γ at this stage in either the binary or the tally representation.

If our algorithm runs in time $2^{w|\phi|}$, then we need time at most $s^{2r}2^{w(ss^q+s^r s^q)}$ to decide whether to put ϕ_e or $\neg\phi_e$ into Γ at this stage in either the binary or the tally representation.

Finally, if necessary, we take the next c_t from our list and write $\delta_s = \theta(c_t)$, which takes additional time $\leq s^q(s^q + 1)$. □

Thus we have the following.

1. If Δ is P-decidable, then the total time required to complete stages 1 through s is of order $\sum_{i=1}^{s} i^k \leq (s^k)^2 = s^{2k}$ for some k.

2. If Δ is $EXPSPACE$-decidable, then the total space required to complete stages 1 through s is of order $s2^{s^k} \leq 2^{s^{2k}}$ for some k.

3. If Δ is $DEXT$-decidable, then the total time required to complete stages 1 through s is of order $s2^{s^k} \leq 2^{s^{2k}}$ for some k.

Claim 3.12 *1. If Δ is P-decidable, then the time required to decide whether $\phi_e \in \Gamma$ is at most $2^{h|\phi_e|}$ for some fixed h.*

2. If Δ is $EXPSPACE$-decidable, then the space required to decide whether $\phi_e \in \Gamma$ is at most $2^{|\phi_e|^h}$ for some fixed h.

3. If Δ is $DEXT$-decidable, then the time required to decide whether $\phi_e \in \Gamma$ is at most $2^{|\phi_e|^h}$ for some fixed h.

Proof: Let $n = |\phi_e|$. By the construction, ϕ_e is decided by stage $s = 2e$. By Claim 3.4, we have $e \leq 2^{cn}$, Thus by Claim 3.11,

1. if Δ is P-decidable, the time required to decide whether ϕ_e is in Γ is at most $\leq (2^{cn})^{2k} \leq 2^{2kcn}$,

2. if Δ is $EXPSPACE$-decidable, the space required to decide whether ϕ_e is in Γ is at most $\leq 2^{(2^{cn})^{2k}} \leq 2^{2^{2kcn}}$, and

3. if Δ is $DEXT$-decidable, the time required to decide whether ϕ_e is in Γ is at most $\leq 2^{(2^{cn})^{2k}} \leq 2^{2^{2kcn}}$.

□

By using a similar analysis as was used in the proof of case (i), we can establish the following.

1. If Δ is P-decidable, then Γ is $DEXT$-decidable and the atomic diagrams of the models $\mathcal{A}$ and $\mathcal{B}$ are in P.
2. If Δ is $EXPSPACE$-decidable, then Γ is $DOUBEXPSPACE$-decidable and the atomic diagrams of the models $\mathcal{A}$ and $\mathcal{B}$ are in $\bigcup_{k>0} DSPACE(2^{n^k})$.
3. If Δ is $DEXT$-decidable, then Γ is $DOUBEXT$-decidable and the atomic diagrams of the models $\mathcal{A}$ and $\mathcal{B}$ are in $EXPTIME$.

This completes the proof of the Main Theorem. $\square$

Corollary 3.13 **(i)** *Any first order theory which is $PSPACE$-decidable in binary (tally) has a complete consistent extension which is $EXPSPACE$-decidable in binary (tally).*

(ii) *Any first order theory which is P-decidable in binary (tally) has a complete consistent extension which is $DEXT$-decidable in binary (tally).*

(iii) *Any first order theory which is $EXPSPACE$-decidable in binary (tally) has a complete consistent extension which is $DOUBEXPSPACE$-decidable in binary (tally).*

(iv) *Any first order theory which is $DEXT$-decidable in binary (tally) has a complete consistent extension which is $DOUBEXT$-decidable in binary (tally).*

Finally, we can modify the proof of the main theorem to include an omitting types argument. Let us say that a non-principal type $\gamma_0(x), \gamma_1(x), \ldots$ in a theory Δ is $PSPACE$-computable in binary (tally) if the following are all true.

(i) For each n, $\Delta \vdash \gamma_{n+1} \rightarrow \gamma_n$

(ii) For each n, $\Delta \nvdash \gamma_n \rightarrow \gamma_{n+1}$;

(iii) There is a $PSPACE$ algorithm which computes γ_n from $bin(n)$ (resp. $tal(n)$) and furthermore $|\gamma_n|$ is bounded by a polynomial in $|bin(n)|$ ($|tal(n)|$).

Similar definitions can be given for other complexity classes and for types over more than one variable. We just give one result here.

Theorem 3.14 *Let Δ be a first order theory which is $PSPACE$-decidable in binary (tally) and let $\{\gamma_n(x)\}_{m\in\omega}$ be a $PSPACE$-computable nonprincipal 1-type. Then Δ has a $PSPACE$-computable model which is $EXPSPACE$-decidable in binary (tally) and which omits the type.*

Proof: We modify the proof of the main theorem as follows. Before starting the construction, we expand Δ^+ to Δ^* by adding $\{\neg\gamma_{n+1}(c_n) : n < \omega\}$. The conditions (i) and (ii) above for a non-principal type imply that Δ^* is still consistent. To see that Δ^* is still $PSPACE$-decidable, observe that for any formula ϕ which contains only constants from $\{c_0, \ldots, c_{n-1}\}$, we have

$$\Delta^* \vdash \phi \longleftrightarrow \Delta^+ \vdash (\neg\gamma_1(c_0) \wedge \cdots \wedge \neg\gamma_n(c_{n-1})) \rightarrow \phi.$$

In fact, the conjunction can be restricted only to those $\neg\gamma_i(c_{i-1})$ for which c_{i-1} occurs in ϕ. Thus we can compute each necessary γ_i from ϕ and the length of the conjunction will be bounded by a polynomial in the length of ϕ, so that we can use the $PSPACE$ procedure for Δ^+ again to decide whether $\Delta^* \vdash \phi$ in space polynomial in $|\phi|$.

□

Similar results of course hold for other complexity classes and for tally as well as binary.

References

[1] Brent, R.: Fast Multiple-Precision Evaluation of Elementary Functions. Journal of the ACM **23** (1976) 242–251.

[2] Cenzer, D., Downey, R. and Remmel, J.B.: Feasible torsion-free Abelian groups. To appear.

[3] Cenzer, D. and Remmel, J.B.: Polynomial-time versus recursive models. Ann. Pure and Appl. Logic **54** (1991) 17–58.

[4] Cenzer, D. and Remmel, J.B.: Polynomial-time Abelian groups. Ann. Pure and Appl. Logic **56** (1992) 313–363.

[5] Cenzer, D. and Remmel, J.B.: Recursively presented games and strategies. Math. Social Sciences **24** (1992) 117–139.

[6] Cenzer, D. and Remmel, J.B.: Feasible graphs and colorings. Math. Logic Quarterly **41** (1995) 327–352.

[7] Cenzer, D. and Remmel, J.B.: Feasibly categorical abelian groups, in *Feasible Mathematics II*, editors P. Clote and J. Remmel, Prog. in Comp. Science and Appl. Logic **13**, Birkhäuser (1995) 91–154.

[8] Cenzer, D. and Remmel, J.B.: Feasibly categorical models, in *Logic and Computational Complexity*, editor D. Leivant, Lecture Notes in Comp. Science **960**, Springer-Verlag(1995) 300–312.

[9] Cenzer, D. and Remmel, J.B.: Feasible graphs with standard universe, in *Computability Theory, Oberwolfach 1996*, Special issue of Ann. Pure and Appl. Logic **94** (1998) 21–35.

[10] Cenzer, D. and Remmel, J.B.: Complexity-theoretic model theory and algebra. *Handbook of Recursive Mathematics, Vol. 1: Recursive Model Theory*, editors Y. Ersov, S. Goncharov, V. Marek, A. Nerode, J. Remmel, Elsevier Studies in Logic and the Foundations of Mathematics, Vol. 138 (1998) 381–513.

[11] Cenzer, D. and Remmel, J.B.: Π_1^0 classes. *Handbook of Recursive Mathematics, Vol. 2: Recursive Algebra, Analysis and Combinatorics*, editors Y. Ersov, S. Goncharov, V. Marek, A. Nerode, J. Remmel, Elsevier Studies in Logic and the Foundations of Mathematics, Vol. 139 (1998) 623–821.

[12] Ebbinghaus, H. and Flum, J.: *Finite Model Theory*, Perspectives in Mathematical Logic, Springer-Verlag (1995).

[13] A. Ehrenfeucht, Separable theories, Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys., 9 (1961) 17-19.

[14] Ferrante, J. and Rackoff, C.: *Complexity of Logical Theories*, Lecture Notes in Mathematics **718**, Springer-Verlag (1979).

[15] Grigorieff, S.: Every Recursive linear ordering has a copy in DTIME(n). J. Symbolic Logic **55** (1990) 260–276.

[16] Harizanov, V.: Pure recursive model theory. *Handbook of Recursive Mathematics, Vol. 1*, Elsevier (1998) 3–114.

[17] Hopcroft, J. and Ullman, J.: "Formal Languages and Their Relations to Automata". Addison Wesley (1969).

[18] C. G. Jockusch and R. I. Soare, Π_1^0 classes and degrees of theories, Trans. Amer. Math. Soc., 173 (1972), 33-56.

[19] Ko, Ker-I: *Complexity theory of real functions,* Progress in Theoretical Computer Science. Birkhauser (1991).

[20] Nerode, A., Remmel, J.B.: Complexity theoretic algebra I, vector spaces over finite fields, *Proceedings of Structure in Complexity, 2d Annual Conference*, Computer Science Press (1987) 218–239.

[21] Nerode, A., Remmel, J.B.: Complexity theoretic algebra II, the free Boolean algebra. Ann. Pure and Applied Logic **44** (1989) 71–99.

[22] Nerode, A., Remmel, J.B.: Complexity theoretic algebra: vector space bases, *Feasible Mathematics*, editors S. Buss and P. Scott, Prog. in Comp. Science and Appl. Logic **9** Birkhäuser (1990) 293–319.

[23] P. Odifreddi: "Classical Recursion Theory". North–Holland, Amsterdam (1989).

[24] C. Papadimitriou: "Computational Complexity". Addison-Wesley (1994).

[25] Remmel, J.B.: When is every recursive linear ordering of type μ recursively isomorphic to a p-time linear order over the binary representation of the natural numbers?, *Feasible Mathematics*, editors S. Buss and P. Scott. Prog. in Comp. Science and Appl. Logic **9** Birkhäuser (1990) 321–341.

[26] Remmel, J.B.: Polynomial-time categoricity and linear orderings, in *Logical Methods*, editors J. Crossley, J. Remmel, R. Shore, and M. Sweedler, Prog. in Comp. Science and Appl. Logic **12** Birkhäuser (1993) 321–341.

[27] Soare, R.: "Recursively enumerable sets and degrees", Springer-Verlag, Berlin (1987).

[28] J. R. Shoenfield, Degrees of models, J. Symbolic Logic, 25 (1960), 233-237.