Undecidability Results for low complexity degree structures
Extended Abstract

Rod Downey[*]
Victoria University of Wellington
New Zealand

André Nies[†]
The University of Chicago
Chicago Illinois 60637
USA

Abstract

We prove that the theory of EXPTIME degrees with respect to polynomial time Turing and many-one reducibility is undecidable. To do so we use a coding method based on ideal lattices of Boolean algebras which was introduced in [7]. The method can be applied in fact to all hyper-polynomial time classes.

1 Introduction

If h is a time constructible function which dominates all polynomials, then, by the methods of the deterministic time hierarchy theorem, $DTIME(h)$ properly contains $\mathcal{P}$. Therefore, a polynomial time reducibility like polynomial time many–one or Turing reducibility induces a nontrivial degree structure on $DTIME(h)$. This degree structure is an uppersemilattice with least element 0. Moreover, by the methods of Ladner ([6], also see [4], Chapter I.7), this degree structure is dense. This was so far the only fact known to hold in general for all such structures. Here we prove that all those degree structures are necessarily complicated, because they have an undecidable first–order theory. In fact, this holds for the degree structure induced on any class of computable languages which contains $DTIME(h)$. Thus, for instance the polynomial T-degrees and many–one degrees of languages in EXPTIME or in $DTIME(2^n)$ have an undecidable theory.

Our results improve previous undecidability results for degree structures in complexity theory, where no reasonable bound on the complexity of the languages involved could be given. Slaman and Shinoda [8] proved that the theory of the polynomial time T-degrees of computable languages is undecidable, and in fact interprets Th($\mathbb{N}$). Ambos–Spies and Nies [3] showed the undecidability of the theory of the polynomial time T-degrees of computable languages. Both proofs make use of the speed–up technique introduced in [1] (which is reminiscent of Blum's speed–up theorem) in order to show that computably presented ideals can be represented as the intersection of two principal ideals. This technique necessarily produces languages of high complexity (usually nonelementary languages).

Most proofs that a problem is undecidable are indirect: one gives a reduction of a problem which is already known to be undecidable to the problem in question. A theory is a consistent set of first order sentences in some language which is closed under logical inference. For theories of structures, a particular type of reduction based on the notion of interpretations of structures is used. It makes use of the following stronger notion of undecidability: call a theory T in a first–order language L *hereditarily undecidable (h.u.)* if each set $X \subseteq T$ which contains the valid L–sentences (i.e. the sentences which can be inferred from $\emptyset$) is undecidable. The transfer principle states that, if $\boldsymbol{A}$ is an L_1–structure, $\boldsymbol{B}$ is an L_2–structure and $\boldsymbol{A}$ can be interpreted in $\boldsymbol{B}$ with parameters, then

$$\mathrm{Th}(\boldsymbol{A}) \text{ h.u. } \Rightarrow \mathrm{Th}(\boldsymbol{B}) \text{ h.u.} \tag{1}$$

See [5], Chapter 5 for a detailed definition of the concept of interpretations of structures. Here we only need the special case that $\boldsymbol{A}$ is a partial order. Then, an interpretation of $\boldsymbol{A}$ in $\boldsymbol{B}$ with a list of parameters $\overline{p}$ is given by formulas $\varphi_U(x;\overline{p})$ and $\varphi_{\leq}(x,y;\overline{p})$ such that,

*Partially supported by the New Zealand Marsden Fund for Basic Science under grant VIC-509

†Partially supported by NSF-grant DMS–9500983 and the New Zealand Marsden Fund for Basic Science under grant VIC-509.

with an appropriate assignment of a list of elements $\overline{b}$ in $\boldsymbol{B}$ to $\overline{p}$, the second formula defines a preordering on $\{c : \boldsymbol{B} \models \varphi_U(c;\overline{b}\}$ so that the partial order obtained by taking the quotient is isomorphic to $\boldsymbol{A}$.

We make use of coding methods developed in [7], where it is shown that intervals of the lattice $\mathcal{E}$ of r.e. languages under inclusion are either boolean algebras or have an undecidable theory. As a tool, in [7] an undecidability result for ideal lattices of certain boolean algebras was proved. Then, an interpretation of such an ideal lattice in intervals of $\mathcal{E}$ is given. Our proof proceeds along the same lines: we give an interpretation of the lattice of Σ^0_2–ideals of an appropriate Σ^0_2-boolean algebra, which satisfies the criteria needed for the auxiliary undecidability result in [7]. By an applications of the transfer principle, this gives the desired undecidability result for our degree structures. The boolean algebra used here is Σ^0_2 because, within a computably presented class $(A_i)_{i\in\omega}$, the question "$A_i \le^p_r A_j$" is Σ^0_2 in i, j.

We assume that all alphabets contain the symbols $0, 1$. For languages $X, Y, X \oplus Y$ denotes the language $0X \cup 1Y$.

2 Σ^0_2– boolean algebras

We give a version of the concepts and the auxiliary result from [7] which is suitable for our use. A Σ^0_2–*boolean algebra* is a boolean algebra $\mathcal{B}$ which can be represented as a model

$$(\mathbb{N}, \preceq, \vee, \wedge)$$

such that $\preceq$ is a Σ^0_2– relation which is a preordering, $\vee, \wedge$ are total computable binary functions, and the quotient structure

$$\mathcal{B} = (\mathbb{N}, \preceq, \vee, \wedge)/_{\equiv}$$

is a boolean algebra (where $n \equiv m \Leftrightarrow n \preceq m \wedge m \preceq n$).

A Σ^0_2–boolean algebra $\mathcal{B}$ is *effectively dense* if there is a computable function F such that

$$x \not\equiv 0 \Rightarrow 0 \prec F(x) \prec x. \tag{2}$$

Without loss of generality we can assume that $\forall x\ F(x) \preceq x$. We will identify sublanguages S of $\mathcal{B}$ with the corresponding preimages $\{n \in \mathbb{N} : n/_{\equiv} \in S\}$. Thus, an ideal of $\mathcal{B}$ is called Σ^0_2 if the preimage is. The Σ^0_2– ideals form a sublattice $\mathcal{I}(\mathcal{B})$ of the distributive lattice of all ideals, because, for Σ^0_2– ideals I, J, the infimum $I \cap J$ and the supremum $I \vee J = \{b \vee c : b \in I \wedge c \in J\}^{\equiv}$ are Σ^0_2 again.

Theorem 2.1 ([7]) *Suppose $\mathcal{B}$ is a Σ^0_2– boolean algebra which is effectively dense. Then $\mathcal{I}(\mathcal{B})$ has a hereditarily undecidable theory.*

Proof. Relativize the proof in [7] of the corresponding result for r.e. boolean algebras to $\emptyset'$ in order to show that $\mathcal{E}^4$ of Σ^0_4–languages under inclusion can be interpreted in $\mathcal{I}(\mathcal{B})$ with parameters. Since $\mathcal{E}^4$ has a h.u. theory, an application of the transfer principle gives the desired result. (This works in fact if the function F in (2) is only Δ^0_2.) $\diamondsuit$

3 Undecidability Results

In the following, let $\le^p_r$ be one of the reducibilities $\le^p_m, \le^p_{1-tt}, \le^p_{btt}, \le^p_{tt}$ or $\le^p_T$. Suppose that $h : \mathbb{N} \mapsto \mathbb{N}$ is an increasing time constructible function with $P \subset DTIME(h)$, so that h eventually dominates all polynomials. $\boldsymbol{D}_r(h)$ denote the degree structure induced by $\le^p_r$ on $DTIME(h)$.

Theorem 3.1 *The elementary theory of $\boldsymbol{D}_r(h)$ is undecidable*

Proof. In a sequence of lemmas, we give an interpretation of $\mathcal{I}(\mathcal{B})$, for an appropriate effectively dense Σ^0_2–boolean algebra $\mathcal{B}$. The plan of the proof is to make $\mathcal{B}$ a very easy, well controlled part of $\boldsymbol{D}_r(h)$, but to use all of $\boldsymbol{D}_r(h)$ to sort out Σ^0_2–ideals of $\mathcal{B}$. We begin with $\mathcal{B}$. For a degree $\boldsymbol{a} \in \boldsymbol{D}_r(h)$, we let $\mathcal{B}(\boldsymbol{a})$ be the set of complemented elements in $[0, a]_{\boldsymbol{D}_r(h)}$, i.e.

$$\mathcal{B}(\boldsymbol{a}) = \{\boldsymbol{x} \le \boldsymbol{a} : \exists \boldsymbol{y}\ \boldsymbol{x} \wedge \boldsymbol{y} = 0\ \wedge\ \boldsymbol{x} \vee \boldsymbol{y} = \boldsymbol{a}\}. \tag{3}$$

We will work with an $\boldsymbol{a}$ which is the r–degree of a set A enjoying the following strong sparseness property introduced by Ambos-Spies.

Definition 3.2 ([1, 2]) A language A is called *super-sparse* if there is a strictly increasing computable function f with domain $\mathbb{N}$, and a procedure M such that

(i) $x \in A$ implies $x = 0^{f(q)}$ for some q.
(ii) $A(0^{f(n)})$ is computable via M in time $f(n+1)+c$.
(iii) $k = f(d)$? is polynomial time in k, d.

Note that a string encoding $\{z : z \in A \wedge |z| \leq f(n)\}$ (e.g. using 1 as a symbol separating between string of 0's) can be computed in time $O(nf(n))$. The property (iii) is called "polynomially honest" in [2]. The numbers k, d are represented in unary.

Ambos-Spies ([2], Theorem 3.2) constructed a supersparse language in $DTIME(2^n)$. His proof works in fact for any class $DTIME(h)$, h as above. The function f is obtained by iterating h, i.e. $f(0) = 1, f(n+1) = h(f(n))$. Thus one obtains

Lemma 3.3 ([2]) *There is a supersparse computable $A \in DTIME(h) - \mathcal{P}$.* $\diamondsuit$

In what follows, we fix such a supersparse A, let $\boldsymbol{a}$ be the r–degree of A and let $\mathcal{B} = \mathcal{B}(\boldsymbol{a})$. A *split* of a language X is a languages B such that for some e, $B = X \cap P_e$. We denote this by $B \sqsubset X$ via P_e. The advantage of taking a supersparse $\boldsymbol{a}$ is that not only is $\mathcal{B}(\boldsymbol{a})$ indeed a boolean algebra, but in fact it is canonically isomorphic to the boolean algebra of splittings of A, modulo the equivalence relation where two splittings are identified if there symmetric difference is in $\mathcal{P}$. The isomorphism is obtained by mapping a split to its degree. In this way, $\mathcal{B}$ is indeed well controlled as desired. (We could in fact easily ensure that A has no infinite $\mathcal{P}$ sublanguages. In that case $\mathcal{B}$ is isomorphic to the boolean algebra of splits modulo finite languages.)

We first show that decomposing A into splits gives complements.

Lemma 3.4 (Ambos-Spies [1]) *Suppose that A is supersparse and $A_1 \sqsubset A$ via P_e. Let $A_2 = A - A_1$. Then the polynomial time T-degrees of A_1 and A_2 form a minimal pair, in the sense that if $Q \leq_T^P A_1, A_2$, then $Q \in P$.*

Proof. Suppose that $Q \leq_T^P A_1, A_2$, with $M_i^{A_i} = Q$ in time $|x|^n$ and $i = 1, 2$. We define a procedure Φ and an auxiliary relation R to compute Q in polynomial time. Inductively, suppose that Φ is correct on all strings of length $\leq s$. For simplicity, as A in not in P, and f is supersparse, we may assume that for all c, $f(c+1) > f(c)^n$. (This assumption can be eliminated by the use of a table look-up for a finite initial segment of A in the definition of Φ below.) For a string of length $s+1$, first see if for some $e \leq s$, $f(e) = (s+1)^n$. If not then let $R(s+1) = R(s)$. If $f(e) = k \leq (s+1)^n$ note that by the assumption above, k is unique. We define $R(s+1) = 2$ if $0^k \in P_e$, and $R(s+1) = 1$ if $0^k \in \overline{P_e}$. Then for q with $|q| = s+1$ define $\Phi(q) = M_{R(s+1)}^{A_{R(s+1)}}(q)$. (Let $r = R(s+1)$. Basically, consider the computation of $M_r^{A_r}(q)$. If an oracle question is has length exceeding $f(e-1)$ then when we query A_r the answer will be 0, since A_r has no elements of length between $f(e-1)$ and $f(e+1)$, so we can simply answer *no* and be correct. If an oracle question has length $\leq f(e-1)$ then as A is supersparse we can decide membership of A on such questions and hence membership of A_r. Thus $\Phi(q)$ can be computed in polynomial time.) $\diamondsuit$

Now we show that, conversely, each pair of complements is represented by a decomposition into splits.

Lemma 3.5 *Suppose that $\boldsymbol{a}_1 \cup \boldsymbol{a}_2 = \boldsymbol{a}$ and $\boldsymbol{a}_1 \cap \boldsymbol{a}_2 = 0$. Then there exists a split $A_1 \sqsubset A$ such that $A_1 \in \boldsymbol{a}_1$ and $A_2 = A - A_1 \in \boldsymbol{a}_2$.*

Proof. It follows from [2], Theorem 4.5, that the polynomial time T-degree of any set $B \leq_T^p A$ collapses to a single 1–tt–degree. Thus it is sufficient to consider the case that $r \in \{m, 1\text{–}tt\}$. It is well known that $\leq_m^p$ and $\leq_{1-tt}^p$ induce distributive uppersemilattices on the computable languages. This is because, if $X \leq_r^p Y \oplus Z$, then there is $P \in \mathcal{P}$ such that $X \cap P \leq_r^p Y$ and $X \cap \overline{P} \leq_r^p Z$ (provided that $r \in \{m, 1\text{–}tt\}$). Now, pick languages $B_i \in \boldsymbol{a}_i$ and apply this to $A \leq_r^p B_1 \oplus B_2$ in order to obtain P. It is sufficient to show that in fact $A_1 = A \cap P \equiv_r^p B_1$ and $A_2 = A \cap \overline{P} \equiv_r^p B_2$. For the first, say, notice that since $B_1 \leq_r^p A_1 \oplus A_2$, there is $Q \in \mathcal{P}$ such that $B_1 \cap Q \leq_r^p A_1$ and $B_1 \cap \overline{Q} \leq_r^p A_2$. But B_1, A_2 form an r– minimal pair, so $B_1 \cap \overline{Q} \in \mathcal{P}$ and therefore $B_1 \equiv_r^p B_1 \cap Q \leq_r^p A_1$. $\diamondsuit$

Finally , we show that the order is preserved when passing from splits to degrees.

Lemma 3.6 *Let $P, Q \in \mathcal{P}$. Then*

$$A \cap P \leq_r^p A \cap Q \Leftrightarrow A \cap (P - Q) \in \mathcal{P}.$$

Proof. The implication from right to left is immediate. For the other implication, notice that $A \cap P$ splits into $A \cap P \cap Q$ and $A \cap (P - Q)$. But $A \cap (P - Q)$ and $A \cap Q$ form a minimal pair by lemma 3.4. Therefore $A \cap (P - Q) \in \mathcal{P}$. $\diamondsuit$

We have obtained a representation of $\mathcal{B}$ in the sense of section 2: Let $e \in \mathbb{N}$ represent $deg_r(A \cap P_e)$. The computable functions $\vee, \wedge$ on $\mathbb{N}$ are obtained by taking unions and intersections of polynomial time languages. Clearly,

"$A \cap P_e \subseteq A \cap P_i$" is Σ^0_2 in e, i.

Lemma 3.7 *$\mathcal{B}$ is effectively dense.*

Proof. By Ladner's uniform diagonalization technique, given a splitting $A \cap P_e$, we can effectively obtain $Q = P_{F(e)} \subseteq P_e$ such that such that $A \cap P_e \notin \mathcal{P}$ implies that $A \cap Q, A \cap (P - Q) \notin \mathcal{P}$ $\diamondsuit$

This concludes our analysis of $\mathcal{B}$. Next we show how to obtain an interpretation of $\mathcal{I}(\mathcal{B})$ in $\boldsymbol{D}_r(h)$. The idea is to represent a Σ^0_2– ideal I by a degree $\boldsymbol{c}_I$ such that

$$I = \{\boldsymbol{b} \in \mathcal{B} : \boldsymbol{b} \leq \boldsymbol{c}_I\}.$$

Clearly any ideal defined in this way must be Σ^0_2 (even if $\boldsymbol{c}_I$ is just the degree of any computable set, not necessarily in $DTIME(h)$). The final lemma will show that, conversely, each Σ^0_2 ideal can be represented in that way by a degree $\boldsymbol{c}_I \in \boldsymbol{D}_r(h)$. Then one obtains the desired interpretation of $\mathcal{B}$ in $\boldsymbol{D}_r(h)$ as follows: the domain formula $\varphi_U(x, a)$ is vacuous (say $x = x$) since each degree represents an ideal. Let

$$\varphi_{\leq}(c_1, c_2; a) \equiv$$
$$\forall x(x \text{ complemented in } [0, a] \Rightarrow (x \leq c_1 \Rightarrow x \leq c_2)).$$

Lemma 3.8 *For each Σ_2 ideal I of $\mathcal{B}$, there exists $C_I \in DTIME(h)$ such that*

$$I = \{e : A \cap P_e \leq^P_r C_I\}.$$

The proof of the following fact is straightforward (see [10]).

Suppose that $q : \mathbb{N} \mapsto \mathbb{N}$ and that $q \leq_T \emptyset'$. The there is a linear time computable $g : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}$ such that for each n, $\lim_s g(n, s) = q(n)$.

Proof of lemma 3.8. If I is a Σ_2 ideal of $\mathcal{B}(A)$ then, since the Σ^0_2 languages are just the ranges of $\emptyset'$-computable functions there exists $q \leq_T \emptyset'$ such that

$$\exists x(q(x) = e) \Leftrightarrow A \cap P_e \in I.$$

By the fact above, we may suppose a linear time computable q such that $\exists x(\lim_s q(x, s) = e)$ iff $A \cap P_e \in I$. We meet the requirements below (where $C = C_I$).

$$\begin{aligned} \mathcal{R}_e : &\quad q(e) = \lim_s q(e, s) \text{ exists} \Rightarrow A \cap P_{q(e)} \leq^p_m C. \\ \mathcal{H}_{\langle e,k \rangle} : &\quad P_k \cap A \neq M^C_e \text{ or } P_k \cap A \in I. \end{aligned}$$

As usual, $\mathcal{H}_{\langle e,k \rangle}$ must live with $\mathcal{R}_p$ for $p \leq \langle e, k \rangle$. At each stage $s \in \mathbb{N}$ we will first compute $h(s)$, using its time constructability, and then perform $h(s)$ steps of the stage s construction below. This ensures that the language $C = C_I$ constructed below is in $DTIME(2h) = DTIME(h)$. We shall call $n \in \mathbb{N}$ *relevant* if it is of the form $n = f(k)$ for some k.

At stage s we do nothing unless s is relevant. In the case that s is relevant, while h permits us (i.e. for $h(s)$ many steps), we consider requirements in decreasing order of priority, beginning with $\mathcal{R}_0$. Once a $\mathcal{H}_j$ requirement has been satisfied (see below) it is no longer considered. Suppose that we are ready to consider requirement $\mathcal{D}$ and this is being done at substage t of stage s.

Case 1. $\mathcal{D} = \mathcal{R}_e$. Let $bin(z)$ denote the binary representation of $z \in \mathbb{N}$. Let $r = \langle e, q(e, s)\rangle$. Define the e-coding location of 0^s to be $c_e(0^s) = bin(r)^\frown 0^{s-|r|}$. Declare that

$$0^s \in A \cap P_{q(e,s)} \text{ iff } c_e(0^s) \in C.$$

Case 2. $\mathcal{D} = \mathcal{H}_{\langle e,k \rangle}$. Let C_t denote what C would be were it the case that stage s were to finish after substage $t-1$. (In particular, no codings will be performed for $\mathcal{R}_j$ for $\mathcal{R}_j$ of lower priority than $\mathcal{H}_{\langle e,k \rangle}$.) Here we are supposing that $h(s)$ is large enough to be able to compute C_t. The key idea is to see if

$$P_k \cap A(0^s) \neq M^{C_t}_e(0^s).$$

If so we declare that $\mathcal{R}_{\langle e,k \rangle}$ is satisfied forever, and finish stage s here. (In effect, initializing all lower priority requirements.)

If $P_k \cap A(0^s) \neq M^{C_t}_e(0^s)$ then do nothing save to move on to substage $t + 1$, ready to consider the next requirement.

It is clear that $C \in DTIME(2h)$ and since a requirement can be initialized only finitely often, since h eventually dominates all polynomials, and since $A \in DTIME(h)$, eventually we consider a requirement $\mathcal{D}$ at almost all stages. Specifically, at each stage s, we first quickly (in linear time) determine if s is relevant. If it is not we move to the next stage. If it is relevant, then we compute $A(0^s)$ which is in $DTIME(h)$. having done this to compute the action of a requirement of the form $\mathcal{R}_e$ requires $q(e,s)$ (linear time) and then $P_e(0^s)$ which needs time s^e. Of course since h dominates $|x|^e$, eventually we have enought time to consider $\mathcal{R}_e$. Similar comments apply to $\mathcal{H}_{\langle e,k\rangle}$.

Since $\lim_s q(e,s) = q(e)$ exists, we see that for almost all n,

$$0^n \in A \cap P_{q(e)} \text{ iff } c_e(0^n) \in C.$$

and hence $A \cap P_{q(e)} \leq_m^P C$.

Finally, suppose that we *fail* to declare that $\mathcal{H}_{\langle e,k\rangle}$ satisfied at any stage. Suppose that s is any stage exceeding a stage s_0 where

(i) all the higher priority $\mathcal{H}$ do ever be declared satisfied have already be so declared,

(ii) for all $j \leq \langle e,k\rangle$, and for all $u \geq s_0$, $q(j,u) = q(j,s_0)$, and

(iii) for all $u \geq s_0$, $h(u)$ is sufficiently large that we have time to consider $\mathcal{H}_{\langle e,k\rangle}$.

The we claim that if $M_e^C = A \cap P_k$, then $A \cap P_k \in I$. Here is the requction. If z is not of the from 0^s or s is not relevant, then $z \notin A \cap P_k$. If $z = 0^s$ and s is relevant, then at the substage t of the stage s construction when we compute C_t, this is computable from $A \cap P_{q(0)}, \ldots, A \cap P_{q(b)}$ where $\mathcal{R}_b$ is the lowest priority $\mathcal{R}$-type requirement of priority exceeding that of $\mathcal{H}_{\langle e,k\rangle}$. (I.e. together with a table from stages $\leq s_0$.) The it can only be that $A \cap P_k(0^s) = M_e^{C_t}(0^s)$. (The reader should note that in the m-reduction case we don't need to compute all of C_t for this step, only the single query to C_t which can be computed in an m-way from $\oplus_{j \leq b} A \cap P_{q(j)}$.)

Hence we se that if we fail to ever declare $\mathcal{H}_{\langle e,k\rangle}$ satisfied, then $A \cap P_k \in I$, since $A \cap P_k \leq \oplus_{j \leq b} A \cap P_{q(j)}$. $\diamondsuit$

References

[1] K. Ambos-Spies, *On the Structure of the Polynomial-time Degrees of Recursive Sets,* Habibitationschrift, Universität Dortmund, 1984.

[2] K. Ambos-Spies, "Inhomogeneities in the Polynomial-time Degrees: the Degrees of Super-Sparse Sets," *Information Processing Letters*, **22** (1986), 113-117.

[3] K. Ambos-Spies and A. Nies "The theory of the polynomial many-one degrees of recursive sets is undecidable," *STACS 92*, Lecture Notes in Computer Science **577** (1992) 209-210.

[4] J. Balcazar, J. Diaz, and J. Gabarro, *Structural Complexity*, Volumes 1 and 2, Springer Verlag (1987,1989).

[5] W. Hodges, *Model Theory*, Encyclopedia of mathematics and its applications 42, Cambridge University Press, 1993.

[6] R. Ladner, "On the Structure of Polynomial Time Reducibility," *J.A.C.M.*, **22** (1975) 155-171.

[7] A. Nies, "Intervals of the lattice of computably enumerable sets and effective boolean algebras," to appear in the J. London Math. Soc.

[8] J. Shinoda and T. Slaman, "On the Theory of PTIME Degrees of Recursive Sets," *J.C.S.S.* **41** (1990) 321-366.

[9] R. Shore and T. Slaman, "The P-T-Degrees of Recursive Sets; Lattice Embeddings, Extensions of Embeddings, and the Two Quantifier Theory," in *Proc. Structures in Complexity* 1989.

[10] R. Soare, *Recursively Enumerable Sets and Degrees*, Springer 1987.