Summary of past research

André Nies

January 31, 2010

Contents

1 Computability and randomness 1
- 1.1 Studying randomness notions . . . 2
  - 1.1.1 Martin-Löf's randomness notion . . . 2
  - 1.1.2 Notions weaker than ML-randomness . . . 2
  - 1.1.3 A notion stronger than ML-randomness . . . 3
- 1.2 Lowness . . . 4
  - 1.2.1 Coincidence of three classes . . . 4
  - 1.2.2 Coincidence with K-triviality . . . 5
  - 1.2.3 Cost functions . . . 6
  - 1.2.4 Strong jump traceability . . . 6

2 Structures from computability theory 7
- 2.1 Coding and definability . . . 7
  - 2.1.1 The Turing degrees of c.e. sets . . . 8
  - 2.1.2 Distributive structures . . . 9
- 2.2 Recent investigations of structures . . . 10

3 Algebra and effectively presented structures 10
- 3.1 Automatic structures . . . 10
- 3.2 Connecting algebra and first-order logic . . . 11

1 Computability and randomness

A mainstay of my research is the interaction between the randomness and the complexity aspects of sets of natural numbers (simply called sets below). The complexity aspect of sets is studied in computability theory. To calibrate the absolute complexity of a set, one introduces classes of similar complexity. Examples of such classes are the computable sets and the low sets (where the halting problem relative to the set as an oracle is as simple as possible). The complexity of a set relative to other sets is measured via reducibilities such as Turing reducibility. The randomness aspect of a set is investigated via computable versions of tools from probability theory and statistics, such as null classes, and tests. Thus, one uses computability theory to introduce a mathematical notion corresponding to the intuitive concept of randomness.

There also is a converse interaction: concepts originating from randomness enrich computability theory. This is the interaction I have stressed in my book [48] and several other publications such as [42]. Also, I devoted to it two of my three invited tutorials at the 2009 summer meeting of the Association for Symbolic Logic in Sofia.

Applying randomness-related notions in computability is the subject of current intense research. In the focus is the now-famous class of K-trivial sets. K-triviality, a property already introduced in 1976 by Chaitin [7], indicates that the set is far from being random. Working with Chaitin, Solovay [52] built a non-computable K-trivial set. Thereafter, the subject lay dormant for over two decades, till it was reconsidered, starting from Solovay's notes, by Calude, Downey, Hirschfeldt and myself from 1999 on.

In the paper [42], I proved the coincidence of K-triviality with several lowness properties (classes of similar complexity that express in some way that a set of natural numbers is close to being computable). For instance, a set A is K-trivial if and only if it is low for Martin-Löf randomness; that is, any random set still appears random when queries to A can be used as an external computational device (oracle) in the effective test notion. This often-cited paper, along with related work carried out with co-workers such as Downey and Hirschfeldt [9, 17], has started a whole new direction of research focusing on the interaction from randomness to computability. The paper also set the template for coincidences of classes. Many further surprising coincidences followed. Classes based on Kolmogorov complexity were shown to coincide with 2-randomness [44, 31]. Lowness properties based on randomness were shown to be equivalent to strong jump traceability [13].

1.1 Studying randomness notions

1.1.1 Martin-Löf's randomness notion

Let 2^ω be the space of infinite binary sequences, and let λ denote the uniform measure on 2^ω where 0,1 both have the probability 1/2. Note that the class $\mathcal{C} \subseteq 2^\omega$ is null if and only if $\mathcal{C} \subseteq \bigcap G_m$ for some sequence $(G_m)_{m\in\mathbb{N}}$ of open sets such that $\lambda G_m \to 0$. We obtain a type of effective null class (or test) by adding *effectivity restrictions* to this condition characterizing null classes. In [29], Martin-Löf introduced a central notion of tests based on computably enumerable objects.

Definition 1.1 A *Martin-Löf test* (or ML-test) is a uniformly computably enumerable sequence $(G_m)_{m\in\mathbb{N}}$ of open sets in 2^ω such that $\lambda G_m \leq 2^{-m}$ for each m.

A set Z is *Martin-Löf random* if Z passes each ML-test, in the sense that Z is not in all of the G_m.

Given a set Z and $n \in \mathbb{N}$, let $Z\restriction_n$ denote the initial segment $Z(0)\ldots Z(n-1)$. Schnorr's Theorem says that Z is ML-random if and only if each initial segment is incompressible with respect to the prefix-free Kolmogorov complexity K.

Theorem 1.2 (Schnorr) *Z is ML-random $\Leftrightarrow$*
there is $b \in \mathbb{N}$ such that $\forall n\, K(Z\restriction_n) \geq n-b$.

A further attractive feature of this notion is that the test concept can be modified in many ways. Its variants lead to a hierarchy of randomness notions. Each one of them formalizes some aspect of our intuition on randomness. For formal definitions and more background on these notions see [48].

1.1.2 Notions weaker than ML-randomness

Schnorr criticized that Martin-Löf-randomness is too strong to be considered algorithmic. He proposed a restricted test notion where we know more about the tests. A *Schnorr test* is a ML-test $(G_m)_{m\in\mathbb{N}}$ such that the measure λG_m is a computable real uniformly in m. A set Z is *Schnorr random* if $Z \notin \bigcap_m G_m$ for each Schnorr test $(G_m)_{m\in\mathbb{N}}$.

Unlike Martin-Löf random sets, Schnorr random sets exist in each high degree by a result with Stephan and Terwijn [44].

Theorem 1.3 *For each set A with $A' \geq_T \emptyset''$, there is a Schnorr random set Z such that $Z \equiv_T A$. If A is computably enumerable then Z can be chosen left-computably enumerable.*

A *randomness enhancement* result has the following form: a weaker randomness notion together with a low complexity property implies a stronger randomness notion. Intuitively speaking, if Z is random in the weaker sense, then being computationally *less* complex implies being *more* random. In [44], with Stephan and Terwijn, I obtained two examples of this for Schnorr randomness.

Theorem 1.4 *(a) Let Z be Schnorr random and not high. Then Z is already ML-random.*
(b) Let Z be Schnorr random and computably dominated. Then Z is already weakly 2-random.

A somewhat mysterious notion implied by Martin-Löf-randomness is Kolmogorov-Loveland (KL) randomness. A main open question in this area is whether the notion actually coincides with ML-randomness. This would defeat Schnorr's critique because KL-randomness is defined using a computable test concept.

Computable randomness is defined by requiring that no computable betting strategies succeeds. Such a strategy places a bet on the next bit position in the usual ascending fashion. We say that Z is *KL random* if no computable betting strategy succeeds even if it is allowed to always choose a next bit position to place a bet. The implications are

$$\text{Martin-Löf random} \Rightarrow \text{KL-random} \Rightarrow \text{computably random} \Rightarrow \text{Schnorr random.}$$

All implications except the leftmost one are known to be strict.

In [30] we obtained various results showing that KL-randomness is, at the very least, much closer to Martin-Löf-randomness than the other notions. The first result says that the restriction of a KL-random set to the even bit positions is already ML-random. For sets A_0, A_1 let $A_0 \oplus A_1$ denote the set $\{2n\colon n \in A_0\} \cup \{2n+1\colon n \in A_1\}$.

Theorem 1.5 *If $A = A_0 \oplus A_1$ is KL-random, then at least one of A_0 and A_1 is Martin-Löf random. If A is Δ^0_2 then in fact both are ML-random.*

Of course this works also for splittings using a very thick co-infinite computable set in place of the even numbers. This shows that the computable dimension of a KL-random set is 1. The main result of the paper extends this to the weaker notion of KL-stochastic sets. Instead of betting on the selected sequence, a test can now only see whether the law of large numbers fails for this sequence.

1.1.3 A notion stronger than ML-randomness

We say that a set Z is *2-random* if Z is Martin-Löf random relative to the halting problem. In [44] we obtained a further randomness enhancement result: a ML-random set is 2-random iff Chaitin's Ω is random relative to it. Further, as a main result we gave a characterization of 2-randomness via plain Kolmogorov complexity C, in the spirit of Schnorr's Theorem. The important difference is that even though one requires incompressibility only for infinitely many initial segments of the set, the incompressibility is taken in the stronger sense of C. One direction was also obtained by Miller [31].

Theorem 1.6 *Z is 2-random $\Leftrightarrow$*
there is $b \in \mathbb{N}$ such that $C(Z\upharpoonright_n) \geq n - b$ for infinitely many n.

As a corollary, we obtained a simple new proof of Kurtz's result [24] that each 2-random set is of hyper-immune degree.

1.2 Lowness

A *lowness property* of a set states that the set is close to being computable. In my book [48], I introduced two paradigms for lowness properties of a set A. In my tutorial at ASL summer meeting, Sofia, 2009, I introduced a third one for Δ^0_2 sets.
Paradigm 1: The set A is not very useful as an oracle. For a formal lowness property of this type, one specifies a sense in which A fails to be useful. Examples are the usual lowness $A' \leq_T \emptyset'$, superlowness $A' \leq_{tt} \emptyset'$, and lowness for randomness (each random set is already random relative to A). Strong jump-traceability is also introduced via this paradigm.
Paradigm 2: the set A is computed by many oracles. For a formal lowness property of this type, one specifies a sense in which the class $\mathcal{S}_A$ of oracles computing A is large (even though $\mathcal{S}_A$ is necessarily a null class for noncomputable A).
Paradigm 3: there is a computable approximation of the set with few changes. The intuition is that the fewer changes one needs in an approximation, the closer the set is to being computable. We will consider this in more detail in Subsection 1.2.3.

Much of my research uses randomness to understand lowness properties. To classify them it is useful to have these paradigms in mind.

1.2.1 Coincidence of three classes

We discuss properties that were introduced independently by various research groups: the low for K sets, the sets that are low for ML-randomness, and the bases for ML-randomness. The first two are examples of Paradigm 1, while the third is Paradigm 2. Later on in Theorem 1.11, we will see that K-triviality can be understood as a lowness property according to Paradigm 3.

(1) In general, adding an oracle A to the computational power of the universal machine decreases $K(y)$. A is *low for K* if this is not so. In other words, $\forall y\, K(y) \leq^+ K^A(y)$. It is not hard to see that such a set is GL_1, namely, $A' \leq_T A \oplus \emptyset'$.

(2) Zambella [53] defined a set A to be *low for ML-randomness* if each ML-random set is already ML-random relative to A. Kučera and Terwijn proved that some non-computable c.e. set is low for ML-randomness [23].

(3) Kučera [22] introduced a further concept expressing computational weakness. We say that A is a *base for ML-randomness* if $A \leq_T Z$ for some $Z \in \mathsf{MLR}^A$. He showed that some non-computable c.e. set is a base for ML-randomness.

Using Schnorr's Theorem and the Kučera-Gacs Theorem, one can easily show that (1)$\Rightarrow$(2) and (2)$\Rightarrow$(3). The following Theorem with Hirschfeldt and Stephan [17] closes the cycle by showing (3)$\Rightarrow$(1).

Theorem 1.7 *Each base for ML-randomness is low for K.*

In the earlier work [42] I showed directly that each set that is low for ML-randomness is low for K. In fact I proved a stronger result. Let CR^A denote the class of computably random sets relative to A. Let MLR denote the ML-random sets.

Theorem 1.8 *If* $\mathsf{MLR} \subseteq \mathsf{CR}^A$ *then* A *is low for* K*.*

This also implies that each set that is low for KL-randomness is low for K. As a further consequence of this theorem and a result in [4], I obtained the following surprising result in [42]:

Theorem 1.9 *Any set that is low for computable randomness is already computable.*

This answers Question 4.8 in Ambos-Spies and Kucera [1] in the negative.

1.2.2 Coincidence with K-triviality

For a string y of length n, up to constants we have $K(0^n) \leq K(y)$, because one can compute n from y. Following Chaitin [7], we say that a set A is K-trivial if, for some $b \in \mathbb{N}$, we also have the converse inequalities: for each n,

$$K(A\restriction_n) \leq K(0^n) + b.$$

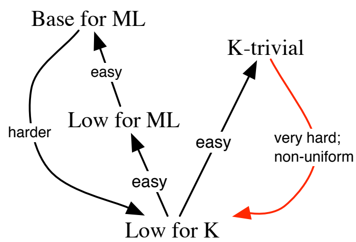
That is, the K complexity of all initial segments is minimal up to a constant. This notion is opposite to ML-randomness by Schnorr's Theorem. Thus, K triviality by itself is not a lowness property but rather expresses being far from random. Chaitin [7] proved that each K-trivial set is Δ^0_2. Solovay [52] showed the existence of a noncomputable K-trivial set. In [9] we found a much simpler injury-free construction of such a set. In fact, we built a c.e. example. To do so, we introduced the cost function construction. We also showed that each K-trivial set is Turing incomplete, using a new method nowadays called the decanter method for of a visualization I introduced (see [10, 48]).

The implication Low for K $\Rightarrow$ K-trivial is easily proved. As a main result in [42], with some input by Hirschfeldt, I proved the converse.

Theorem 1.10 *Each* K*-trivial set is low for* K*.*

The proof method is now known as the golden run method. It extends the decanter method. It is necessarily non-uniform [42]: the constant for being low for K cannot be computed from an index for the given (say, c.e.) set together with the constant for being K-trivial. A golden run is a node in a tree of possible runs of procedures which shows that the set is low for K via a specific constant.

The following diagram summarizes the implications discussed:



1.2.3 Cost functions

The theory of cost functions began in [9, 42, 48]. Cost functions can be used to analyse subclasses of the K-trivial sets.

A cost function is a computable function that maps a pair of natural numbers x, s to a nonnegative rational $c(x, s)$. When building a Δ^0_2 set A, the number $c(x, s)$ is interpreted as the cost of changing the computable approximation to the bit $A(x)$ at stage s. We say that A *obeys* the cost function if for some computable approximation; the sum of the cost of changes is finite. This constitutes the third lowness paradigm: A is close to computable because the total of changes, measured by a cost function, can be made small.

Many cost functions are introduced via randomness-related concepts. For instance, the so-called standard cost function $c_{\mathcal{K}}$ is based on the prefix-free Kolmogorov complexity $K(w)$ of a string w, which may be viewed as the binary representation of a number:

$$c_{\mathcal{K}}(x, s) = \textstyle\sum_{w=x+1}^{s} 2^{-K_s(w)},$$

where $K_s(w)$ is the value of $K(w)$ obtained by stage s. In [42], using the golden run method, I characterized the K-trivial sets with a single cost function.

Theorem 1.11 *A set A obeys the standard cost function if and only if A is K-trivial.*

As a corollary, I showed that K-triviality is essentially a property of c.e. Turing degrees.

Theorem 1.12 *Each K-trivial set A is Turing below a computably enumerable K-trivial set D.*

To prove this one thinks of $\mathbb{N}$ as partitioned into infinitely many columns: The column for x is $\{\langle x, i\rangle \colon i \in \mathbb{N}\}$ with the standard pairing function $\langle \cdot, \cdot \rangle$. The set D is simply the "change set" for a computably approximation of A. Each time $A(x)$ changes, one enumerates the next element in the column for x into A. Clearly $A \leq_T D$. Since A obeys the standard cost function, it is not hard to see that D obeys the standard cost function as well, and hence is K-trivial.

1.2.4 Strong jump traceability

The strongly jump traceable (SJT) sets were introduced in [12]. They have been in the focus of intense research in the past few years. For instance, the Advances in Maths paper [8]) shows that the c.e. strongly jump traceable sets form a proper subclass of the c.e. K-trivials.

Strong jump traceability is a lowness property according to Paradigm 1. The intuition is that the universal A-partial computable functional J^A has only very few possible values. For the formal definition, a *c.e. trace* for a partial function ψ is a uniformly c.e. sequence $(T_x)_{x \in \mathbb{N}}$ such that for all $x \in \mathrm{dom}(\psi)$ we have $\psi(x) \in T_x$. An *order function* is a computable, nondecreasing, and unbounded function $h \colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$. A c.e. trace $(T_x)_{x \in \mathbb{N}}$ is *bounded* by an order function h if for all x, $|T_x| \leq h(x)$. A set A is *strongly jump traceable* if for every order function h, J^A has a c.e. trace that is bounded by h.

Since the order function h is arbitrary, this is an extreme lowness property. Nonetheless, in [12] we show that there is some non-computable c.e. set in this class. Further, we give a characterization in terms of relativized plain Kolmogorov complexity. Similar to the property of being low for K, the characterization states that adding A as an oracle does not reduce the complexity of a string by much.

Theorem 1.13 *A is strongly jump traceable iff for each order function g, we have* $\forall x\,[C(x) \leq C^A(x) + g(C^A(x)) + O(1)]$.

We say that A is *strongly superlow* if for each order function h, one can computably approximate its jump A' with the number of changes bounded by h. Using the foregoing theorem, we showed that each strongly superlow set is strongly jump traceable.

A result of Greenberg and Nies [14] analogous to Theorem 1.11 characterizes strong jump traceability in terms of obeying all members in a family of well-behaved cost functions c (called benign): the number of disjoint intervals $[x, s)$ such that $c(x, s) \geq 1/n$ is bounded by a computable function $g(n)$. This shows that cost functions can enable us to carry out an analytical treatment of lowness properties inside the Δ^0_2 sets.

2 Structures from computability theory

A main goal of computability theory is to understand the computational complexity of sets of natural numbers. The relative complexity of a set is given by comparing it to other sets via a reducibility $\leq_r$. Such reducibilities specify a way in which the oracle B can be accessed in a relative computation of A. Important examples are Turing reducibility $A \leq_T B$, where full access to the oracle is granted, weak truth-table reducibility $A \leq_{\mathrm{wtt}} B$, where the largest oracle question is computably bounded in the input, and many-one reducibiltiy $A \leq_m B$, where from the input x a single oracle question $f(x)$ is computed, and the output is simply $B(f(x))$.

A reducibility $\leq_r$ is a pre-ordering $\leq_r$ on the power set of $\mathbb{N}$, and thus induces a partial ordering on equivalence classes, called the *degree structure* given by $\leq_r$. In this way, one abstracts from the particular features of a set A, and only studies its relative complexity as given by $\leq_r$. The degree structures form uppersemilattices where the supremum of the degrees of sets A, B is the degree of $A \oplus B$.

A somewhat purer view of the computably enumerable sets is given by $\mathcal{E}$, the distributive lattice of computably enumerable sets under inclusion. Surprisingly, the behaviour of a set within $\mathcal{E}$ can tell us a lot about its computational complexity. For instance, by a result of Martin [28], maximal sets are high, and each high computably enumerable degree contains a maximal set.

2.1 Coding and definability

The study of structures from computability via coding with first-order formulas was in the focus my research during the 1990s. A simple example of a such a coding is to represent a symmetric graph (V, E) in a suitable partial order. The vertices in V correspond to the minimal elements. Whenever there is an edge between vertices v and w, there is an element r above v and w. Thus, the domain is defined by the formula saying that an element is minimal, and the edge relation E by the formula $\phi(v, w) \equiv \exists r[v \leq r \;\&\; w \leq r]$.

If one codes a finite graph, say, in a given degree structure, one has to find a *coding scheme*, consisting of a pair of formulas to define the relations corresponding to V and E. These formulas will involve *parameters*, that is, constants taken from the degree structure. Methods specific to the degree structure in question (for instance, the priority method explained below) will then be used to construct the right parameters.

The set representing the domain V usually becomes a parameter definable antichain in the degree structure. Often the degree structure is dense, so one cannot take minimal elements; more complex coding schemes become necessary.

In my habilitation thesis [46], a fair part of my research in this direction is summarized and put into a wider context. The thesis also stresses the application of coding methods to structures from computational complexity theory, such as the polynomial time many-one degrees of exponential time sets.

2.1.1 The Turing degrees of c.e. sets

The analysis via coding and definability was highly successful in the case of a central structure: the structure $\mathcal{R}_T$ of Turing degrees of computably enumerable sets. This analysis began with the result of Harrington and Shelah [16] that the first-order theory $\mathrm{Th}(\mathcal{R}_T)$ is undecidable. Extending the coding methods used there, Harrington and Slaman (unpublished) gave an interpretation in $\mathrm{Th}(\mathcal{R}_T)$ of $\mathrm{Th}(\mathbb{N}, +, \times)$, also called true arithmetic. (An *interpretation* is a many-one-reduction of theories based on a computable map defined in some natural way on sentences.) With Shore and Slaman [43], I used codings of standard models of arithmetic in order to prove the definability in $\mathcal{R}_T$ of important degree classes.

Theorem 2.1 ([43]) *The classes* Low_2 *and* $High_1$ *are first-order definable without parameters in* $\mathcal{R}_T$.

We also coded $(\mathbb{N}, +, \times)$ into $\mathcal{R}_T$ using no parameters at all.

A main open question is whether $\mathcal{R}_T$ has a non-trivial automorphism. We showed in the same paper [43] that no automorphism of $\mathcal{R}_T$ can change the second jump of a degree. In [34] I used coding methods to show a further restricting result: each automorphism of $\mathcal{R}_T$ is arithmetical (in fact Σ^0_7) on each proper final segment. To do so I gave a definable mapping from a coded standard model onto any such final segment. This also shows that the ideal generated by a parameterless-definable set is itself parameterless-definable.

In order to strengthen an undecidability result for the theory of a structure, besides interpreting true arithmetic one can look at the level of quantifier alternations where the theory becomes undecidable. To lower this level, codings with very simple formulas are needed. In this direction, with Lempp and Slaman I proved the following.

Theorem 2.2 ([26]) *The* $\forall\exists\forall$*-theory of* $\mathcal{R}_T$ *(as a partial order) is undecidable.*

We combined the Harrington-Shelah type of coding with algebraic tricks to reduce the quantifier alternations: the coding merely uses Σ_1 formulas with parameters. Later on, with Miller and Shore [32], I strengthened this by showing that the Π_2 theory of $(\mathcal{R}_T, \vee, \wedge)$ is undecidable (where $\wedge$ is denotes some total extension of the partial meet operator). We coded with quantifier free formulas in this language.

All these results combine coding via first-order formulas with the priority method. Originating independently in work of Friedberg and Muchnik in the 1950s, the priority method was almost synonymous with computability theory in the 1980s and 1990s. In somewhat modified form, it persists in the more recent investigations where randomness and computability interact.

The priority method makes it possible to simultaneously pursue strategies for different requirements, even if they have conflicting goals. For this reason, the method can be seen as an abstract version of computing with many processors. Complicated control devices, such as trees of strategies, were developed to resolve the conflicts between strategies. The above-mentioned results [43, 26, 32] rely on sophisticated applications of the priority method at the $\emptyset''$ and even $\emptyset'''$ levels. For instance, an antichain in the computably enumerable Turing degrees can be defined by taking the minimal degrees $\mathbf{g} \le \mathbf{r}$ such that $\mathbf{q} \le \mathbf{p} \vee \mathbf{g}$. A priority construction is necessary to build parameters $\mathbf{p}, \mathbf{q}, \mathbf{r}$ so that one obtains an infinite antichain. This defined antichain can then serve as the domain of a coded standard model of arithmetic.

2.1.2 Distributive structures

We say that an upper semilattice $(U, \le, \vee)$ is *distributive* if for $x, y, z \in U$, if $z \le x \vee y$ then z splits into components $z_x \le x$ and $z_y \le y$, namely, $z = z_x \vee z_y$. For lattices, this coincides with the usual concept of distributivity. Important examples of distributive upper semilattices are the degree structures $\mathcal{R}_{wtt}$ and $\mathcal{R}_m$ of weak truth-table and of many-one degrees on computably enumerable sets, respectively.

Much of my earlier work focuses on coding into distributive structures from computability theory. This property of well-behavedness restricts the availability of coding schemes. For instance, the scheme mentioned above to define antichains in $\mathcal{R}_T$ will not work in distributive structures.

In an early paper [35], I proved the following.

Theorem 2.3 *True arithmetic can be interpreted in the theory of $\mathcal{R}_m$. In fact, $\mathcal{R}_m$ allows a parameter-free coding of $(\mathbb{N}, +, \times)$.*

This paved the way for the solution by Harrington and Nies [15] of a long-standing problem posed, for instance, in [51].

Theorem 2.4 *True arithmetic can be interpreted in the theory of $\mathcal{E}$, the distributive lattice of computably enumerable sets.*

The idea taken from [35] was to use recursion on k in order to give first-order definitions with parameters of relations with a Σ^0_k index set. One starts with simple relations (usually when $k = 3$) and then works ones way up to more complicated relations by recursion to the simpler case.

The coding into $\mathcal{E}$ is very indirect. To this day, the lowest level where the theory is known to become undecidable it Π_6, and in fact this only holds for the closely related lattice $\mathcal{E}^*$ which is the quotient of $\mathcal{E}$ by finite differences of sets. This result was obtained in [46]. Further, no infinite linear order can be interpreted in any structure of the form $A \times A$ by a result of Hodges and Nies [20]. Since $\mathcal{E}$ is isomorphic to $\mathcal{E} \times \mathcal{E}$, the analog of the second part of Theorem 2.3 fails.

A computably enumerable Boolean algebra $\mathcal{B}$ is called *effectively dense* if from (an index for) a nonzero element one can compute a nonzero element that is strictly below it. For instance, the Lindenbaum algebra of sentences modulo Peano arithmetic is effectively dense. Let $\mathcal{I}(\mathcal{B})$ be the lattice of computably enumerable ideals of $\mathcal{B}$. I transferred the results and methods on coding in $\mathcal{E}$ to the lattices $\mathcal{I}(\mathcal{B})$ for an effectively dense Boolean algebra $\mathcal{B}$. I first showed undecidability of its theory [37]. Thereafter, I proceeded to an interpretation of true arithmetic [38].

The point is that structures $\mathcal{I}(\mathcal{B})$ can often be interpreted without much effort in other distributive structures. In the same paper with Downey [11], I applied this method to structures that are studied in computational complexity theory, a branch of theoretical computer science.

Theorem 2.5 *The polynomial time many-one and Turing degrees of exponential time sets have an undecidable theory.*

A well-known open problem was solved in my paper [2], with my advisor Ambos-Spies and Shore: the theory of $\mathcal{R}_{wtt}$, the weak truth table degrees of c.e. sets, is undecidable. Strengthening this, in [39] true arithmetic was interpreted in the theory of $\mathcal{R}_{wtt}$.

Turning to the computably enumerable many-one degrees $\mathcal{R}_m$ [36] , I obtained Π_3 as a level where the theory of the partial order becomes undecidable. This relied on my general method for showing undecidability of fragments, embodied by the *transfer lemma* [36]. First I showed hereditary undecidability of the Π_3 theory of the class of finite distributive lattices in the language of partial orders. This undecidability result can then be transferred to the Π_3 theory of $\mathcal{R}_m$ by a theorem of Lachlan that each finite distributive lattice is isomorphic to an initial segment of $\mathcal{R}_m$.

Lempp and Nies [25] applied the transfer lemma from [36] in order to show undecidability of the Π_4-theory of $\mathcal{R}_{wtt}$ as a partial order. Unlike $\mathcal{R}_m$, the degree structure $\mathcal{R}_{wtt}$ is dense, which explains the increase by one quantifier alternation.

2.2 Recent investigations of structures

Prompted in part by considerations related to randomness, I have recently worked again on structures from computability theory.

In [27], with Lewis and Sorbi I prove:

Theorem 2.6 *The first-order theory of both the Medvedev and the Muchnik lattices are equivalent in complexity to third-order arithmetic.*

In [3], Barmpalias and I show the following.

Theorem 2.7 *Each proper Σ^0_3 ideal of the computably enumerable Turing degrees is bounded by a low$_2$ c.e. degree. Each proper Σ^0_4 ideal of the computably enumerable Turing degrees is bounded by an incomplete c.e. degree.*

The first result gives a low$_2$ upper bound in $\mathcal{R}_T$ for the K-trivial degrees. The second answered a question of Calhoun [6] by showing that no proper prime ideal of $\mathcal{R}_T$ is Σ^0_4.

3 Algebra and effectively presented structures

My paper "Describing groups" [47] surveys two directions of research related to algebra, which I developed in several research papers.

3.1 Automatic structures

Automata are used to represent algebraic structures. For instance, the integers with addition can be represented in that way. Such representations are of interest in theoretical computer science because algebraic structures can be viewed as an abstract model for data structures, and a representation by automata is extremely efficient.

This line of research yielded the publications [21, 18] with various co-authors at the 2004 and 2008 LICS conferences. One main result in [21] is the following theorem, which shows that despite the apparently very strong restriction due to the representability by finite automata, these structures form in general a very rich class.

Theorem 3.1 *Isomorphism between presentations of automatic graphs is undecidable, and in fact Σ^1_1 complete.*

We also give in [21] a complete characterization of the finite automata presentable Boolean algebras. In this case, the class is indeed very restricted. Let $\mathcal{B}$ be the Boolean algebra of finite or cofinite subsets of $\mathbb{N}$. Clearly $\mathcal{B}$ is finite automata presentable.

Theorem 3.2 *An infinite Boolean algebra is finite automata presentable if and only if it is isomorphic to a finite power of* $\mathcal{B}$.

With Thomas, I proved strong restrictions on finite automata presentable groups.

Theorem 3.3 ([45]) *Let G be a finite automata presentable infinite group. Then each finitely generated subgroup H of G is abelian-by-finite.*

Via some matrix theory, the preceding theorem also leads to a strong restriction on finite automata presentable rings.

Theorem 3.4 ([45]) *Let R be a finite automata presentable ring (possibly non-commutative). Then R is locally finite.*

This implies that the only finite automata presentable rings (commutative or not) without zero divisors are the finite fields.

In [18] we consider structures of size the continuum that can be represented via Büchi automata. We refute a claim made in [5], by showing that there is a Büchi presentable structure without an injective Büchi presentation. We derive this from a stronger result obtained with methods from descriptive set theory.

Theorem 3.5 *There a is Büchi presentable structure without an injective Borel presentation.*

The theory of Borel structures has been further developed in a recent paper with Hjorth [19]. Its main result shows that the completeness theorem has no effective version for uncountable structures, when effectivity for uncountable structures is interpreted by being Borel.

Theorem 3.6 *There is a complete Borel theory without a Borel model.*

3.2 Connecting algebra and first-order logic

The second direction surveyed in [47] combines algebra and first-order logic. It began with my paper "Separating classes of groups by first order sentences" [41]. Logicians at Univ. Paris 7 (Sabbagh, Oger and others) have been interested in the topic over the past few years. This resulted in two papers in the Journal of Group Theory [50, 49]. Further, the model theorists Scanlon (UC Berkeley) and Aschenbrenner (UCLA) have worked in this direction.

The main goal is to understand how expressive first-order logic is within the context of groups. In [41] several important classes of groups are separated via the first-order theory. For instance:

Theorem 3.7 *There is a sentence that holds in all finitely presented groups, but fails in some finitely generated group.*

To prove such theorems I introduced the following very fruitful concept. A finitely generated group G is called *quasi-finitely axiomatizable* if it satisfies a certain first order sentence, and each finitely generated group also satisfying that sentence is isomorphic to G. Thus, G can be axiomatized by a single sentence within the class of finitely generated groups. I gave several examples, such as the Heisenberg group $\mathrm{UT}_3^3(\mathbb{Z})$. Oger [49] showed that some groups originating from number theory are quasi-finitely axiomatizable. Oger and Sabbagh [50, 49] gave an algebraic characterization of being quasi-finitely axiomatizable for nilpotent groups.

The property of being quasi-finitely axiomatizable is conceptually very close to being a prime model. Howevery, there is an important difference. Clearly there are only countably many quasi-finitely axiomatizable groups. On the other hand, in [33] I showed:

Theorem 3.8 *There are uncountably many non-isomorphic finitely generated groups that are prime models of their theories.*

Thus, not each prime finitely generated group is quasi-finitely axiomatizable. It remains open whether each quasi-finitely axiomatizable group is prime.

My paper [40] contains further results connecting algebra and first-order logic. Let F_2 be the free group of rank 2. I show that F_2 is ω-homogeneous. In fact:

Theorem 3.9 *Any two tuples of the same length in F_2 satisfying the same existential formulas are automorphic.*

I also show that the theory of F_2 has no prime model.

References

[1] K. Ambos-Spies and A. Kučera. Randomness in computability theory. In Peter Cholak, Steffen Lempp, Manny Lerman, and Richard Shore, editors, *Computability Theory and Its Applications: Current Trends and Open Problems*. American Mathematical Society, 2000.

[2] K. Ambos-Spies, A. Nies, and R. Shore. The theory of the recursively enumerable weak truth-table degrees is undecidable. *J. Symbolic Logic*, 57(3):864–874, 1992.

[3] G. Barmpalias and A. Nies. Upper bounds for ideals in the computably enumerable degrees. Submitted, 20xx.

[4] B. Bedregal and A. Nies. Lowness properties of reals and hyper-immunity. In *Wollic 2003*, volume 84 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2003. Available at `http://www.elsevier.nl/locate/entcs/volume84.html`.

[5] Achim Blumensath and Erich Grädel. Finite presentations of infinite structures: automata and interpretations. *Theory Comput. Syst.*, 37(6):641–674, 2004.

[6] W. Calhoun. Incomparable prime ideals of the recursively enumerable degrees. *Ann. Pure Appl. Logic*, 63:36–56, 1993.

[7] G. Chaitin. Information-theoretical characterizations of recursive infinite strings. *Theoretical Computer Science*, 2:45–48, 1976.

[8] P. Cholak, R. Downey, and N. Greenberg. Strongly jump-traceability I: the computably enumerable case. *Adv. in Math.*, 217:2045–2074, 2008.

[9] R. Downey, D. Hirschfeldt, A. Nies, and F. Stephan. Trivial reals. In *Proceedings of the 7th and 8th Asian Logic Conferences*, pages 103–131, Singapore, 2003. Singapore Univ. Press.

[10] R. Downey, D. Hirschfeldt, A. Nies, and S. Terwijn. Calibrating randomness. *Bull. Symbolic Logic*, 12(3):411–491, 2006.

[11] R. Downey and A. Nies. Undecidability results for low complexity degree structures. In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity*, pages 128–132, 1997. Extended version to appear in *Journ. Comp. Sys. Sc.*

[12] S. Figueira, A. Nies, and F. Stephan. Lowness properties and approximations of the jump. *Ann. Pure Appl. Logic*, 152:51–66, 2008.

[13] N. Greenberg, D. Hirschfeldt, and A. Nies. Characterizing the strongly jump traceable sets via randomness. To appear.

[14] N. Greenberg and A. Nies. Benign cost functions and lowness properties. To appear.

[15] L. A. Harrington and A. Nies. Coding in the lattice of enumerable sets. *Adv. in Math.*, 133:133–162, 1998.

[16] Leo A. Harrington and S. Shelah. The undecidability of the recursively enumerable degrees. *Bull. Amer. Math. Soc.*, 6:79–80, 1982.

[17] D. Hirschfeldt, A. Nies, and F. Stephan. Using random sets as oracles. *J. Lond. Math. Soc. (2)*, 75(3):610–622, 2007.

[18] G. Hjorth, B. Khoussainov, A. Montalban, and A. Nies. From automatic structures to borel structures. In *Proceedings of the 19th IEEE Symposium on Logic in Computer Science*, Lecture Notes in Computer Science, pages 110–119. IEEE Computer Society, 2008.

[19] Greg Hjorth and Andre Nies. Borel structures and Borel theories. To appear in JSL.

[20] W. Hodges and A. Nies. Interpreting infinite linear orders. In J.A. Makowsky and E.V. Ravve, editors, *Logic Colloquium '95*, volume 11 of *Lecture Notes in Logic*, pages 247–251, 1997.

[21] B. Khoussainov, A. Nies, S. Rubin, and F. Stephan. Automatic structures: richness and limitations. In *Proceedings of the 19th IEEE Symposium on Logic in Computer Science*, Lecture Notes in Computer Science, pages 110–119. IEEE Computer Society, 2004.

[22] A. Kučera. On relative randomness. *Ann. Pure Appl. Logic*, 63:61–67, 1993.

[23] A. Kučera and S. Terwijn. Lowness for the class of random sets. *J. Symbolic Logic*, 64:1396–1402, 1999.

[24] S. Kurtz. *Randomness and genericity in the degrees of unsolvability.* Ph.D. Dissertation, University of Illinois, Urbana, 1981.

[25] S. Lempp and A Nies. Undecidability of the 4-quantifier theory for the recursively enumerable turing and wtt degrees. *J. Symbolic Logic*, 60:1118–35, 1995.

[26] Steffen Lempp, André Nies, and Theodore A. Slaman. The π_3-theory of the enumerable Turing degrees is undecidable. *Trans. Amer. Math. Soc.*, 350(7):2719–2736, 1998.

[27] A. Lewis, A. Nies, and A. Sorbi. The first order theories of the Medvedev and Muchnik lattices. In *5th Conference on Computability in Europe, Cie 2009*, volume LNCS 5635 of *Lecture Notes in Computer Science*, pages 324–331, Berlin, Heidelberg, 2009. Springer–Verlag.

[28] D. A. Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Z. Math. Logik Grundlag. Math.*, 12:295–310, 1966.

[29] P. Martin-Löf. The definition of random sequences. *Inform. and Control*, 9:602–619, 1966.

[30] W. Merkle, J. Miller, A. Nies, J. Reimann, and F. Stephan. Kolmogorov-Loveland randomness and stochasticity. *Ann. Pure Appl. Logic*, 138(1-3):183–210, 2006.

[31] J. Miller. Every 2-random real is Kolmogorov random. *J. Symbolic Logic*, 69:907–913, 2004.

[32] R. Miller, A. Nies, and R. Shore. The $\forall\exists$ theory of $(R, \vee, \wedge)$ is undecidable. *Trans. Amer. Math. Soc.*, 356(8):3025–3067, 2004.

[33] A. Nies. Comparing quasi-finitely axiomatizable groups and prime groups. To appear.

[34] A. Nies. Parameter definable subsets of the recursively enumerable degrees. To appear.

[35] A. Nies. The last question on recursively enumerable many-one degrees. *Algebra i Logika*, 33(5):550–563, 1995. English Translation July 1995.

[36] A Nies. Undecidable fragments of elementary theories. *Algebra Universalis*, 35:8–33, 1996.

[37] A. Nies. Intervals of the lattice of computably enumerable sets and effective boolean algebras. *Bull. Lond. Math. Soc.*, 29:683–92, 1997.

[38] A. Nies. Effectively dense boolean algebras and their applications. *Trans. Amer. Math. Soc.*, 352(11):4989–5012, 2000.

[39] A. Nies. Interpreting $\mathbb{N}$ in the c.e. weak truth table degrees. *Ann. Pure. Appl. Logic*, 107:35–48, 2001.

[40] A. Nies. Aspects of free groups. *J. Algebra*, 263:119–125, 2003.

[41] A. Nies. Separating classes of groups by first–order formulas. *Intern. J. Algebra Computation*, 13:287–302, 2003.

[42] A. Nies. Lowness properties and randomness. *Adv. in Math.*, 197:274–305, 2005.

[43] A. Nies, R. Shore, and T. Slaman. Interpretability and definability in the recursively enumerable Turing-degrees. *Proc. Lond. Math. Soc.*, 3(77):241–291, 1998.

[44] A. Nies, F. Stephan, and S. Terwijn. Randomness, relativization and Turing degrees. *J. Symbolic Logic*, 70(2):515–535, 2005.

[45] A. Nies and R. Thomas. FA-presented groups and rings. To appear in J. Algebra.

[46] André Nies. Coding methods in computability theory and complexity theory, 1998. Habilitationsschrift, Universität Heidelberg.

[47] André Nies. Describing groups. *Bull. Symbolic Logic*, 13(3):305–339, 2007.

[48] André Nies. *Computability and randomness*, volume 51 of *Oxford Logic Guides*. Oxford University Press, Oxford, 2009.

[49] F. Oger. Quasi-finitely axiomatizable groups and groups which are prime models. *J. Group Theory*, 9(1):107–116, 2006.

[50] F. Oger and G. Sabbagh. Quasi-finitely axiomatizable nilpotent groups. *J. Group Theory*, 9(1):95–106, 2006.

[51] R. Soare. *Recursively Enumerable Sets and Degrees.* Perspectives in Mathematical Logic, Omega Series. Springer–Verlag, Heidelberg, 1987.

[52] R. Solovay. Handwritten manuscript related to Chaitin's work. IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 215 pages, 1975.

[53] D. Zambella. On sequences with simple initial segments. ILLC technical report ML 1990-05, Univ. Amsterdam, 1990.