Direct Construction of Scott Ideals

Russell Miller[1,2](✉)

[1] Queens College, 65-30 Kissena Blvd., Queens, NY 11367, U.S.A.
Russell.Miller@qc.cuny.edu
[2] C.U.N.Y. Graduate Center, 365 Fifth Avenue, New York, NY 10016, U.S.A.

Abstract. A Scott ideal is an ideal $\mathcal{I}$ in the Turing degrees, closed downwards and under join, such that for every degree $\boldsymbol{d}$ in $\mathcal{I}$, there is another degree in $\mathcal{I}$ that is a PA-degree relative to $\boldsymbol{d}$. It is known that, for every Turing degree $\boldsymbol{d}$, there is a Scott ideal containing $\boldsymbol{d}$ in which every degree is low relative to $\boldsymbol{d}$ (with jump Turing-reducible to $\boldsymbol{d}'$). We give a construction of such an ideal $\mathcal{I}_{\boldsymbol{d}}$, uniform in a given set $D \in \boldsymbol{d}$, using the Uniform Low Basis Theorem of Brattka, de Brecht, and Pauly. The primary contribution of this article may be the questions posed at the end about the monotonicity of this construction.

Keywords: computability theory · PA-degree · Π^0_1-class · Scott ideal · Turing degree · Turing reducibility · Uniform Low Basis Theorem

1 Introduction

Finite-branching trees are ubiquitous in mathematical logic. They arise as soon as one begins to consider the completions of a consistent theory. For a consistent, decidable axiom set, such as the axioms **PA** for Peano arithmetic, the complete consistent extensions of the axiom set correspond bijectively to the (infinite) paths through a decidable subtree of the complete binary tree $2^{<\omega}$.

From the Incompleteness Theorem of Gödel, we immediately realize that the subtree above, despite all its decidability, has no computable path. Kreisel remarked that the Halting Problem $\emptyset'$ must be able to compute some path through each such tree. Shoenfield went further in [8], proving that every such tree contains a path whose Turing degree lies strictly below the degree $\mathbf{0}'$ of $\emptyset'$. But it was Jockusch and Soare who claimed the sharpest result, as their Low Basis Theorem in [3] established that every such tree has an infinite path of *low* Turing degree $\boldsymbol{d}$, i.e., with jump $\boldsymbol{d}' = \mathbf{0}'$. Such a path is "almost" computable, in the sense that its relativization of the Halting Problem is no more complicated than the actual Halting Problem.

The paths through the tree described above for the axiom set **PA** came to be called the ***PA**-degrees*, the Turing degrees of complete consistent extensions of

R. Miller—The author was partially supported by Grant #581896 from the Simons Foundation and by the City University of New York PSC-CUNY Research Award Program. The composition of this article was aided by useful conversations with Emma Dinowitz.

G. Della Vedova et al. (Eds.): CiE 2023, LNCS 13967, pp. 23–34, 2023.
https://doi.org/10.1007/978-3-031-36978-0_3

PA. It turns out that these are precisely the degrees capable of computing some path through every decidable infinite subtree of $2^{<\omega}$. In turn, the term "**PA** degree" was relativized: for an arbitrary degree $\boldsymbol{a}$, a degree $\boldsymbol{d}$ is ***PA** relative to* $\boldsymbol{a}$ if $\boldsymbol{d}$ computes a path through every $\boldsymbol{a}$-computable infinite subtree of $2^{<\omega}$. (Equivalently, $\boldsymbol{d}$ computes a path through every $\boldsymbol{a}$-computable infinite finite-branching tree whose branching function is $\boldsymbol{a}$-computable.) The results of [3] relativize to show that for every Turing degree $\boldsymbol{a}$, there is a degree $\boldsymbol{d}$ that is both **PA** relative to $\boldsymbol{a}$ and low relative to $\boldsymbol{a}$, that is, with $\boldsymbol{a}' = \boldsymbol{d}'$. (Necessarily $\boldsymbol{a} < \boldsymbol{d}$ for every $\boldsymbol{d}$ that is **PA** relative to $\boldsymbol{a}$. Here the relations $<$ and $\leq$ on degrees always denote Turing reducibility.)

More recently, in 2012, Brattka, de Brecht, and Pauly gave a uniform version of the Low Basis Theorem, constructing a Turing functional Γ such that, for every $A \subseteq \omega$, $\Gamma^A(n, s)$ computes a function on ω^2 whose limit, as $s \to \infty$, is the characteristic function of the jump of a set D of **PA**-degree relative to A. Thus D itself must be low relative to A, as A' allows one to compute D'.

Our purpose in this abstract is to apply the Uniform Low Basis Theorem (Theorem 2 below) to give a uniform construction of Scott ideals, a well-known concept in reverse mathematics that we introduce in Sect. 3. As we will explain in Sect. 4, however, our ultimate goal is not related to reverse math, but rather to the absolute Galois group of the rational numbers. In light of this goal, two open questions naturally arise, to be described (but not answered) in that section. We view these questions as important and challenging. Quite possibly the questions themselves are the most important items in this abstract. The technical constructions preceding them require attention but follow a predictable path and lead to results that will not seem foreign or unusual.

2 Constructing PA Degrees Uniformly

For the degree $\mathbf{0}$, the tree described above, whose paths are the complete extensions of **PA**, has the property that the degrees of paths through that tree are precisely the **PA** degrees relative to $\mathbf{0}$. The first result here, which is well-known, relativizes this statement to an arbitrary degree $\boldsymbol{a}$ and says that such a tree can be created uniformly in a set $A \in \boldsymbol{a}$.

Theorem 1. *There is a computable relation $R \subseteq \omega \times 2^{<\omega} \times 2^\omega$ such that, for every $A \in 2^\omega$, the set*

$$\{\sigma \in 2^{<\omega} : (\forall m, n \leq |\sigma|) R(n, \sigma\restriction m, A)\}$$

forms an A-computable subtree $T_A \subseteq 2^{<\omega}$ such that, for every degree $\boldsymbol{c}$,

$$\boldsymbol{c} \text{ is PA relative to } A \iff \boldsymbol{c} \text{ computes a path through } T_A.$$

Moreover, since T_A is computable uniformly from A, there is a computable total injective function $h : \omega \to \omega$ such that, for every A, h is a 1-reduction from the jump $(T_A)'$ to the jump A'.

T_A is built so that its paths are the consistent completions of the axiom set **PA** augmented by axioms saying $f = \chi_A$, in the language of **PA** with a unary function symbol f adjoined. If Υ is the Turing functional such that $T_A = \Upsilon^A$, then the function h, on input e, outputs the code number of the Turing functional $\Phi_{h(e)}$ such that

$$\Phi^C_{h(e)}(x) = \Phi^{\Upsilon^C}_e(e).$$

Thus, for arbitrary sets A,

$$h(e) \in A' \iff \Phi^A_{h(e)}(h(e))\downarrow \iff \Phi^{\Upsilon^A}_e(e)\downarrow \iff \Phi^{T_A}_e(e)\downarrow \iff e \in (T_A)'.$$

Theorem 2 (Uniform Low Basis Theorem: Thm. 8.3 in [1]). *There exists a Turing functional Γ such that, for every set $A \subseteq \omega$, Γ^A is total and there exists a set P_A, of **PA** degree relative to A, such that*

$$(\forall n)\ \lim_s \Gamma^A(n,s) = \chi_{(P_A)'}(n),$$

The set P_A may be viewed as a path through the universal A-computable subtree T_A of $2^{<\omega}$. This path is *low relative to* T, meaning that $(P_A)' \leq_T A'$, as its jump $(P_A)'$ is the limit of an A-computable function. The original Low Basis Theorem of Jockusch and Soare [3], relativized to A, proved the existence of such a path. Brattka, de Brecht, and Pauly showed that the jump of P_A can be approximated uniformly using a A-oracle.

With this, for an arbitrary set A, we will define an infinite sequence $A = A_0 <_T A_1 <_T A_2 <_T \cdots$ of subsets of ω such that:

- For every $n \in \omega$, deg A_{n+1} is a PA degree relative to A_n; and
- There exists an A-computable function M, which we will call a *master function*, such that

$$(\forall n)(\forall x)\ \lim_{s\to\infty} M(n,x,s) = \begin{cases} 1, \text{ if } x \in A'_n \\ 0, \text{ if } x \notin A'_n. \end{cases}$$

The first condition automatically implies $A_{n+1} \not\leq_T A_n$. The second condition yields a uniform-limit result for the sets A_n themselves, using the following easy lemma.

Lemma 1. *There is a computable total function f such that, for every $C \subseteq \omega$, f is a 1-reduction from C to C'.*

Proof. Define $f(n)$ to be the index e of the functional Φ_e given by

$$\Phi^B_e(x) = \begin{cases} 0, \text{ if } n \in B; \\ \uparrow, \text{ if not.} \end{cases}$$ □

Thus, uniformly for all n and x, $A_n(x) = \lim_s M(n, f(x), s)$. However, the uniform computation of the jumps A'_n is substantially stronger than this. One might say that the sets A_n are *uniformly low*, as their jumps are uniformly limit-computable in A (or equivalently, uniformly A'-computable).

Our sequence is readily defined. It begins with $A_0 = A$. Next, given A_n, we define T_n to be the tree T_{A_n} as given in Theorem 1, which shows that T_n may be computed uniformly from A_n. Then Theorem 2 yields a path $P_{n+1} = P_{T_n}$ through this tree T_n. By Theorem 1, the Turing degree of P_{n+1} is a PA degree relative to A_n, and we define $A_{n+1} = P_{n+1}$.

The sequence $\{A_n\}_{n\in\omega}$ will instantiate the following Proposition.

Proposition 1. *For arbitrary $A \subseteq \omega$, there exists a strictly ascending sequence $\{A_n\}_{n\in\omega}$ of subsets of ω, all low relative to A, with $A_0 = A$ and such that every A_{n+1} has PA degree relative to A_n, and an A'-computable master function M such that $M(n, x) = \chi_{A'_n}(x)$ for all n and x.*

Proof. We use the sets A_n defined above. Since A_{n+1} is a PA degree relative to A_n, we immediately have $A_n <_T A_{n+1}$. (The reduction $A_n \leq_T A_{n+1}$ follows by considering a single fixed index e such that Φ_e^B defines the subtree of $2^{<\omega}$ whose nodes are just the initial segments of B. The path P_{n+1} computes a path through the tree Φ_e^A, hence computes A_n; and since the same index e works for every n, this reduction is uniform in n.)

Next we show how to compute the required master function $M(n, x)$. Of course, since $A_0 = A$, we begin with $M(0, x) = \chi_{A'}(x)$. Assuming by induction on n that, with the A'-oracle, we have computed $M(n, x) = \chi_{A'_n}(x)$ for all x, we now address $(A_{n+1})' = (P_{n+1})'$. Recall from Theorem 1 that for the tree $T_n = T_{A_n}$, there is a computable 1-reduction h from $(T_n)'$ to $(A_n)'$. Moreover, by Theorem 2, the jump $(P_{n+1})' = (P_{T_n})'$ is given by $\lim_s \Gamma^{T_n}(x, s)$. Therefore

$$x \in (A_{n+1})' \iff \lim_s \Gamma^{T_n}(x, s) = 1.$$

We search for a number s_0 such that $(\forall s \geq s_0)\Gamma^{T_n}(x, s) = \Gamma^{T_n}(x, s_0)$. For each s_0, this is a question about the membership in $(T_n)'$ of a particular index $g(s_0)$ (computable from s_0 uniformly in x and n). The function h allows us to convert this question into the question of membership of $h(g(s_0))$ in $(A_n)'$, which we can compute with our oracle, by inductive hypothesis. Moreover, $\lim_s \Gamma^{T_n}(x, s)$ exists, so eventually we find such an s_0, and when we do, we define $M(n, x) = \Gamma^{T_n}(x, s_0)$. Thus $M(n + 1, x) = \chi_{(A_{n+1})'}(x)$ as desired. □

3 Defining Scott Ideals

Proposition 1 allows us to build Scott ideals uniformly. We recall the relevant definition. (Certain applications of it will be discussed briefly in Sect. 4.)

Definition 1. *A nonempty set $\mathcal{I}$ of Turing degrees is a* Turing ideal *if it is closed downward under Turing reducibility $\leq$ and also under the finite join operation. A Turing ideal $\mathcal{I}$ is a* Scott ideal *if it has the additional property that, for every $\boldsymbol{a} \in \mathcal{I}$, $\mathcal{I}$ also contains a $\boldsymbol{PA}$-degree relative to $\boldsymbol{a}$.*

The most common Turing ideals are the *principal ideals* $\{\boldsymbol{d} : \boldsymbol{d} \leq \boldsymbol{c}\}$ defined by any single degree $\boldsymbol{c}$. These are not Scott ideals, however, as they contain $\boldsymbol{c}$

but no degree $> \boldsymbol{c}$ and consequently no degree **PA** relative to $\boldsymbol{c}$. The natural strategy for building a Scott ideal $\mathcal{I}$ (especially if one wants $\mathcal{I}$ to be countable) is to start with a degree $\boldsymbol{a}$ and close under the condition of Definition 1, adjoining to $\mathcal{I}$ some degree $\boldsymbol{a}_1$ **PA** relative to $\boldsymbol{a}$, then another degree $\boldsymbol{a}_2$ **PA** relative to $\boldsymbol{a}_1$, and so on. (Of course, when adjoining $\boldsymbol{a}_n$, we also adjoin all degrees $\leq \boldsymbol{a}_n$.) Proposition 1 shows exactly how to do this effectively, while keeping each $\boldsymbol{a}_n$ as small in the Turing hierarchy as possible. Indeed, since each $\boldsymbol{a}_{n+1}$ is low relative to $\boldsymbol{a}_n$, we have $\boldsymbol{a}'_{n+1} \leq \boldsymbol{a}'_n \leq \cdots \leq \boldsymbol{a}'_1 \leq \boldsymbol{a}'$, so the entire ideal stays very close to the starting point $\boldsymbol{a}$.

The point of the upcoming Theorem 4 is to give a cleaner definition of the Scott ideal than the foregoing. The concept of an *exact pair* of Turing degrees appears in Theorem 3 in Spector's article [12], which in turn cites the article [4] by Kleene and Post. Rephrased in the language of the textbook [11], it states that every strictly ascending sequence of Turing degrees (under Turing reducibility $\leq$) has an exact pair.

Theorem 3 (Kleene-Post-Spector; see Theorem VI.4.2 in [11]). *For every sequence $\{\boldsymbol{a}_n\}_{n\in\omega}$ of Turing degrees with $\boldsymbol{a}_n < \boldsymbol{a}_{n+1}$ for all n, there exist upper bounds $\boldsymbol{b}$ and $\boldsymbol{c}$ (called an* exact pair *for the sequence) such that*

$$(\forall \boldsymbol{d})\ [[\boldsymbol{d} \leq \boldsymbol{b}\ \&\ \boldsymbol{d} \leq \boldsymbol{c}] \iff [\exists n\ \boldsymbol{d} \leq \boldsymbol{a}_n]].$$

We wish to apply Theorem 3 to the sequence of degrees $\boldsymbol{a}_n$ of the sets A_n produced by Proposition 1 above, taking advantage of the specific properties of that sequence. Recall that every degree $\boldsymbol{a}_n$ from that sequence is low relative to the given set A of degree $\boldsymbol{a}$, with $\boldsymbol{a}'_n = \boldsymbol{a}'$. Moreover, we have a single uniform computable approximation of their jumps. In Proposition 1, the master function was given as an A'-computable function $M(n, x)$. Here we use the equivalent formulation of an A-computable master function $M(n, x, s)$, with $\chi_{(A_n)'}(x) = \lim_{s\to\infty} M(n, x, s)$ for all n and x. (This is better adapted to our construction in Theorem 4 below, whereas the A'-computable version simplified the proof of Proposition 1.) On the other hand, every A_{n+1} computes a path through the universal strongly-A_n-computable subtree T_n of $2^{<\omega}$, and so $A_{n+1} \not\leq_T A_n$. We noted earlier that $A_n \leq_T A_{n+1}$ uniformly in n, so the sequence of degrees $\{\boldsymbol{a}_n\}$ is indeed strictly ascending, allowing us to apply the following theorem to it.

Theorem 4 (Low exact pairs for uniform low ascending sequences). *For every strictly ascending sequence $\{A_n\}_{n\in\omega}$ of subsets of ω such that the reductions $A_n \leq_T A_{n+1}$ are computable uniformly in n, and for every master function M with $\lim_{s\to\infty} M(n, x, s) = \chi_{A'_n}(x)$ for all n and x, there exist subsets B and C of ω that form an exact pair for the sequence $\{A_n\}$ and whose join is low relative to M (i.e., $(B \oplus C)' \leq_T M'$).*

Proof. We give the proof under the assumption that A_0 and M are computable. Relativizing to an arbitrary A_0 and an A_0-computable M is trivial. In our construction, $S =^* T$ denotes that the symmetric difference of the sets S and T is finite, while $S^{[n]} = \{m \in \omega : \langle n, m\rangle \in S\}$ is the n-th "column" of the set S, when the subset S of ω is viewed as a two-dimensional array using a computable bijection $\langle\cdot,\cdot\rangle$ from ω^2 onto ω. This and other standard notation comes from [11], in which Theorem VI.4.2 is a non-effective version of the proof given here. We use requirements similar to those there, for all i, j, and n, along with our lowness requirements for all e:

$$\begin{aligned} \mathcal{T}_n^B &: B^{[n]} =^* A_n. \\ \mathcal{T}_n^C &: C^{[n]} =^* A_n. \\ \mathcal{L}_e &: \text{if } (\exists^\infty s)\ \Phi_{e,s}^{\beta_s\oplus\gamma_s}(e)\downarrow\text{, then } \Phi_e^{B\oplus C}(e)\downarrow. \\ \mathcal{R}_{\langle i,j\rangle} &: \text{if } \Phi_i^B = \Phi_j^C \text{ and both are total, then } (\exists n)\ \Phi_i^B \leq_T A_n. \end{aligned}$$

In the last of these, β_s and γ_s are the s-th finite strings in the computable approximations $\{\beta_s\}_{s\in\omega}$ and $\{\gamma_s\}_{s\in\omega}$ that we will build, with $B = \lim_s \beta_s$ and $C = \lim_s \gamma_s$. It is well known that, if all of these $\mathcal{L}$-requirements are satisfied, then $(B \oplus C)$ will indeed be a low set. The $\mathcal{R}$-requirements will establish that every set computable both from B and from C will be computable from some A_n, while the $\mathcal{T}$-requirements yield the converse, that every A_n is both B-computable and C-computable. Each $\mathcal{T}_n^B$ is given priority over $\mathcal{T}_n^C$, which has priority over $\mathcal{L}_n$, which has priority over $\mathcal{R}_n$, and all of these have priority over $\mathcal{T}_{n+1}^B$.

(The original result of Spector, Theorem 3 in [12], is somewhat effective, noting that both B and C lie strictly below the jump $(\oplus_n A_n)'$ of the infinite join of the sets A_n. This also holds of the construction in [11], although it goes unmentioned there. The construction of [11] is somewhat easier to imitate, so we adapt it here. The new result here in Theorem 4 is the lowness of $(B \oplus C)$, which will follow from the new assumption of uniform lowness of the sets A_n.)

As is common, each requirement $\mathcal{L}_e$ may impose a *restraint* $l(e,s)$ at each stage $s \geq e$. This will mean that only requirements of higher priority than $\mathcal{L}_e$ may move elements $< l(e,s)$ into or out of $(\beta_s \oplus \gamma_s)$ at stage $s+1$. Likewise, a requirement $\mathcal{R}_k = \mathcal{R}_{\langle i,j\rangle}$ may impose a restraint $r(k,s)$ at that stage, which must be similarly respected by all lower-priority requirements. The restraint $l(e,s)$ will help ensure that $\Phi_e^{B\oplus C}(e)\downarrow$, once convergence has occurred at a finite stage. The restraint $r(k,s)$ will help preserve computations $\Phi_i^B(x)\downarrow\neq \Phi_j^C(x)\downarrow$, once they have been seen to occur at a finite stage.

At stage 0, all restraints are set with $l(e,0) = r(k,0) = 0$, and β_0 and γ_0 are both the empty string. It is convenient to consider every requirement to be *initialized* at this stage. (Each time a requirement is injured, it will be re-initialized.)

At each stage $s+1$ of the construction, we first consider the $\mathcal{L}$-requirements. For each $e \leq s$, we check whether the computation $\Phi_{e,s}^{\beta_s\oplus\gamma_s}(e)$ halts. If it does halt, let u be the use of this computation (i.e., the greatest cell on the oracle tape that is read by the machine during the computation), and set $l(e,s+1) = \lfloor \frac{u+1}{2} \rfloor$. This will cause our procedure to protect the first u bits of $\beta_s \oplus \gamma_s$, with priority e. We do this independently for every $e \leq s$; newly chosen restraints are *not* considered to have injured lower-priority requirements, so no requirements are initialized at this point.

Next we check which $\mathcal{R}$-requirements *need attention*. To determine whether $\mathcal{R}_k$ needs attention at stage $s+1$, with $k = \langle i, j \rangle$, let e_{ks} be the code number of a program which uses an A_n-oracle to search for strings σ and τ in $2^{<\omega}$ and $t, y \in \omega$ such that all of the following hold.

1. For all $k' \leq k$ and all $\langle k', m \rangle$ with $r(k', s) \leq \langle k', m \rangle < |\sigma|$, $\sigma(\langle k', m \rangle) = A_{k'}(m)$. (Notice that checking this requires an A_k-oracle, along with the uniform reductions $A_{k'} \leq_T A_k$ for all $k' < k$.)
2. For all $k' \leq k$ and all $\langle k', m \rangle$ with $r(k', s) \leq \langle k', m \rangle < |\tau|$, $\tau(\langle k', m \rangle) = A_{k'}(m)$. (Again this requires an A_k-oracle.)
3. $(\forall e \leq k)(\forall x < l(e, s+1))\ [\sigma(x) = \beta_s(x)\ \&\ \tau(x) = \gamma_s(x)]$.
4. $(\forall k' < k)(\forall x < r(k', s))\ [\sigma(x) = \beta_s(x)\ \&\ \tau(x) = \gamma_s(x)]$.
5. $\Phi^{\sigma}_{i,t}(y)\!\downarrow\ \neq \Phi^{\tau}_{j,t}(y)\!\downarrow$.

The first of these says that setting $\beta_{s+1} = \sigma$ would not injure the higher-priority $\mathcal{T}_B$ requirements, and the next one says that setting $\gamma_{s+1} = \tau$ would not injure the higher-priority $\mathcal{T}_C$ requirements. Items (3) and (4) say that doing this would not injure any higher-priority $\mathcal{L}$- or $\mathcal{R}$-requirements, and the last item says that doing it would satisfy $\mathcal{R}_k$ (provided $\beta_{s+1} \oplus \gamma_{s+1}$ is preserved thereafter).

If $M(n, e_{ks}, s) = 0$, then our master function currently guesses that there are no such σ and τ, and so $\mathcal{R}_k$ does not need attention at stage $s+1$. Also, if $\mathcal{R}_k$ has received attention at a previous stage and has not been injured since that stage, then it does not need attention now. (In this case, taking $\sigma = \beta_s$ and $\tau = \gamma_s$ satisfies all these conditions!) Otherwise, with $M(n, e_{ks}, s) = 1$, we search either until we find (σ, τ, y, t) satisfying all of these conditions, or until we reach a stage $s' > s$ at which $M(n, e_{ks}, s') = 0$. If we first find such a stage s', then the master function's current guess is later superseded and $\mathcal{R}_k$ does not need attention at stage $s+1$; but if we first find the tuple (σ, τ, y, t), then it does need attention. (Notice that one or the other of these must occur, because if $\lim_{s'} M(n, e_{ks}, s') = 1$, then $e_{ks} \in (A_n)'$, meaning that the search must terminate with the discovery of a tuple (σ, τ, y, t).)

If there is no $k \leq s$ for which $\mathcal{R}_k$ needs attention at this stage, then for the pair $\langle n, m \rangle = |\beta_s|$, we define $\beta_{s+1} = \beta_s{}^\frown(A_n(m))$, extending β_s by a single bit which is either 1 or 0 according to whether $m \in A_n$ or not. Thus this last bit agrees with the requirement T^B_n. Likewise, for $\langle n', m' \rangle = |\gamma_s|$, we define $\gamma_{s+1} = \gamma_s{}^\frown(A_{n'}(m'))$, as demanded by $T^C_{n'}$. In this case all restraints are preserved: $l(e, s+1) = l(e, s)$ and $r(k, s+1) = r(k, s)$.

If there exists a $k = \langle i, j \rangle \leq s$ such that $\mathcal{R}_k$ needs attention at stage $s+1$, then for the least such k, we find the (least) tuple (σ, τ, y, t) that satisfies all five conditions and define

$$\beta_{s+1}(x) = \begin{cases} \sigma(x), & \text{if } x < |\sigma|; \\ \beta_s(x), & \text{if } |\sigma| \leq x < |\beta_s|; \\ A_n(m), & \text{if } |\sigma| \leq |\beta_s| = x = \langle n, m \rangle \end{cases}$$

and

$$\gamma_{s+1}(x) = \begin{cases} \tau(x), & \text{if } x < |\tau|; \\ \gamma_s(x), & \text{if } |\tau| \leq x < |\gamma_s|; \\ A_n(m), & \text{if } |\tau| \leq |\gamma_s| = x = \langle n, m \rangle. \end{cases}$$

Thus $\sigma \sqsubseteq \beta_{s+1}$ and $\tau \sqsubseteq \gamma_{s+1}$, and we have filled in bits as needed to ensure $|\beta_{s+1}| > |\beta_s|$ and $|\gamma_{s+1}| > |\gamma_s|$. For every $e > k$, both $\mathcal{R}_e$ and $\mathcal{L}_e$ are initialized at this stage, with $l(e, s+1) = r(e, s+1) = 0$. We keep $l(e, s+1) = l(e, s)$ for all $e \leq k$ and $r(e, s+1) = r(e, s)$ for every $e < k$. For $\mathcal{R}_k$ itself, we define $r(k, s+1)= \max(|\sigma|, |\tau|)$, which is sufficiently long to protect both of the computations $\Phi_{i,t}^{\beta_{s+1}}(y)$ and $\Phi_{j,t}^{\gamma_{s+1}}(y)$ in Condition (5). $\mathcal{R}_k$ is said to have *received attention* at this stage.

This completes stage $s+1$. We will define $B = \lim_s \beta_s$ and $C = \lim_s \gamma_s$ once we have shown that these limits actually exist. To see that $\lim_s \beta_s$ exists, observe first that in our construction, $|\beta_{s+1}| > |\beta_s|$ for all s, and second that the only situation in the construction that can cause an incompatibility in these strings (with $\beta_{s+1} \not\sqsupseteq \beta_s$) is when a requirement $\mathcal{R}_k$ receives attention. Fix an arbitrary $x \in \omega$ and consider the first stage s_0 at which some requirement $\mathcal{R}_{k_0} = \mathcal{R}_{\langle i,j \rangle}$ causes $\beta_{s_0-1}(x) \neq \beta_{s_0}(x)$. At this stage, $\mathcal{R}_{k_0}$ becomes satisfied; the only reason why it might act again is if it is subsequently injured, and the only requirements that can injure it are higher-priority $\mathcal{R}$-requirements (as no other requirements can alter $\beta_s \restriction r(k, s_0)$ or $\gamma_s \restriction r(k, s_0)$). But when a higher-priority requirement $\mathcal{R}_{k_1}$ acts at a stage $s_1 > s_0$, it defines β_{s_1} with length $> |\beta_{s_0}|$. Therefore, after stage s_1, only requirements of higher priority than $\mathcal{R}_{k_1}$ can redefine $\beta_s(x)$. Continuing by induction, we see that $\beta_s(x)$ may be redefined at most k_0-many times after stage s_0, so eventually it stabilizes. Thus $\lim_s \beta_s(x)$ always exists, as does $\lim_s \gamma_s(x)$ by the same argument, so B and C are indeed well-defined.

It remains to show that this B and C satisfy our requirements and consequently instantiate the theorem. We argue by induction on the priority of the requirements, starting with $\mathcal{T}_0^B$ and proving that each one is satisfied, that it only receives attention at finitely many stages, and that any relevant restraints $l(e, s)$ or $r(k, s)$ stabilize at finite values as $s \to \infty$.

For a requirement $\mathcal{T}_n^B$, notice first that whenever β_s is extended by one bit to β_{s+1} (at stages $s+1$ at which no $\mathcal{R}$-requirement needs attention), the new bit is always defined to satisfy the (only) relevant $\mathcal{T}^B$-requirement. The only other way that bits in β_{s+1} can be redefined or newly defined is if a requirement $\mathcal{R}_k$ receives attention. By inductive hypothesis there is a stage s_0 after which no higher-priority requirement acts again. But if $k \geq n$, then Condition (4) ensures that the relevant string σ must match β_s on $\omega^{[n]} \cap \{0, 1, \ldots, r(n, s) - 1\}$, while Condition (1) ensures that σ must match A_n on the rest of the column $\omega^{[n]}$ (up to $|\sigma|$). Thus, after stage s_0, no further extension of β_s will cause any more disagreements between A_n and $B^{[n]}$, and no bit in $\beta_s \restriction \omega^{[n]}$ will ever be redefined again once it has entered $\mathrm{dom}(\beta_s)$ for some s. Thus $\mathcal{T}_n^B$ is satisfied. A parallel argument shows that $\mathcal{T}_n^C$ is also satisfied.

For the requirement $\mathcal{L}_e$, we again fix a stage s_0 after which no higher-priority requirement than $\mathcal{L}_e$ receives attention. Assume that there are indeed infinitely many stages s at which $\Phi_{e,s}^{\beta_s \oplus \gamma_s}(e) \downarrow$, and fix the first such stage $s_1 > s_0$. The construction therefore sets $l(e, s_1 + 1) = \lfloor \frac{u+1}{2} \rfloor$, where u is the use of the computation $\Phi_{e,s_1}^{\beta_{s_1} \oplus \gamma_{s_1}}(e)$. Only $\mathcal{R}$-requirements could possibly cause $\beta_s \restriction l(e, s_1) \neq \beta_{s_1} \restriction l(e, s_1)$ at subsequent stages s, and by inductive hypothesis no higher-priority $\mathcal{R}$-requirements ever do so. But whenever an $\mathcal{R}_k$ with $k \geq e$ redefines $\beta_{s+1} \oplus \gamma_{s+1}$ to equal some new $\sigma \oplus \tau$ incompatible with $\beta_s \oplus \gamma_s$, Condition (3) forces it to choose them so that $\sigma \upharpoonright l(e, s+1) = \beta_s \upharpoonright l(e, s+1)$ and $\tau \upharpoonright l(e, s+1) = \gamma_s \upharpoonright l(e, s+1)$. Therefore the computation $\Phi_{e,s_1}^{\beta_{s_1} \oplus \gamma_{s_1}}(e)$ is preserved at all subsequent stages, and so $\Phi_e^{B \oplus C}(e)\downarrow$, satisfying $\mathcal{L}_e$. Moreover, $\mathcal{L}_e$ never again redefines $l(e, s+1)$, as the current $l(e, s_1+1)$ preserves the convergence.

Finally, consider a requirement $\mathcal{R}_k = \mathcal{R}_{\langle i,j \rangle}$, and suppose that $\Phi_i^B = \Phi_j^C = f$ is a total function from ω to $\{0, 1\}$. We fix a stage s_0 after which no higher-priority requirement acts again, and give a program Ψ^{A_k} for computing f from an A_k-oracle. (Here k is in fact the index $\langle i, j \rangle$ of the requirement.) Ψ^{A_k} begins with finitely much information: the stage s_0 and the strings β_{s_0} and γ_{s_0}. On input y, it searches for either a string σ or a string τ that satisfy all of the Conditions (1)-(4) (as listed on page 7) relevant to itself and such that either $\Phi_i^\sigma(y)$ or $\Phi_j^\tau(y)$ converges. (For a σ, Condition (2) is irrelevant, as is (1) for a τ; also, only half of (3) and (4) is relevant to either.) Whichever of those two programs halts, Ψ^{A_k} outputs that value, claiming that it must be the value of $f(y)$. (Notice that we did need the oracle A_k here, in order to verify satisfaction of Conditions (1) and (2).)

We remark first that Ψ^{A_k} does indeed compute a total function. After all, $\Phi_i^B(y)$ halts, using some finite initial segment $B \upharpoonright u$ of its oracle B, as does $\Phi_j^C(y)$ using some $C \upharpoonright u$. But every extension of $\beta_{s_0} \oplus \gamma_{s_0}$ to a subsequent $\beta_s \oplus \gamma_s$ in our construction must have satisfied all of Conditions (1)-(4), because no requirement of lower priority than $\mathcal{R}_k$ ever acted again. (One-bit extensions, at stages when no $\mathcal{R}_{k'}$ needed attention, also satisfy those conditions.) Therefore, if it does not find some other σ or τ first, Ψ^{A_k} will eventually find some $\beta_s \supseteq B \upharpoonright u$ or some $\gamma_s \supseteq C \upharpoonright u$ which satisfy the conditions, and it will output $\Phi_i^B(y)$ or $\Phi_j^C(y)$ accordingly. In this case, those are both the correct value $f(y)$ that we wanted it to output. It remains to show that, even if it found some other σ or τ first, Ψ^{A_k} still outputs the correct value $f(y)$.

So suppose that there is some σ (say) which satisfied all the relevant conditions and had $\Phi_i^\sigma(y)\downarrow$, thus producing our value $\Psi^{A_k}(y)$. If $\Psi^{A_k}(y) \neq f(y)$, then $\Phi_i^\sigma(y) \neq \Phi_j^C(y)$. Let $C \upharpoonright u$ be the initial segment used here, and find an $s_1 > s_0$ such that $\gamma_s \upharpoonright u = C \upharpoonright u$ for all $s \geq s_1$. We claim now that at all stages $s+1 > s_1$, $\mathcal{R}_k$'s need for attention was witnessed by some finite σ_1 extending the given σ, by $\tau = \gamma_{s+1}$, and by the given y and $t = s$. Indeed $\Phi_i^{\sigma_1}(y)\downarrow \neq \Phi_j^\tau(y)\downarrow$, satisfying (5), and the satisfaction of all the other conditions follows from their satisfaction by σ and by γ_{s+1}. Since this held at all stages $> s_1$, the program $\Phi_{e_{ks}}^{A_k}$ halts on every input, making $e_{ks} \in (A_n)'$. But $M(n, e_{ks}, t)$ approximates $(A_k)'$, so at a sufficiently large stage t it will have $M(n, e_{ks}, t) = 1$, and at this stage $\mathcal{R}_k$ needed attention. By the choice of s_0, no higher-priority $R_{k'}$ needed attention at that stage, so $\mathcal{R}_k$ will have received attention at that stage, with $\beta_{t+1} \supseteq \sigma_1$ and $\gamma_{t+1} \supseteq \tau$. Since $\mathcal{R}_k$ is never injured again, the halting computations $\Phi_i^{\beta_{t+1}}(y)$ and $\Phi_j^{\gamma_{t+1}}(y)$ will have been preserved forever after, contradicting the hypothesis that $\Phi_i^B = \Phi_j^C$. A symmetric argument shows that no τ can have caused $\Psi^{A_k}(y)$

to output an incorrect value, so indeed $f = \Psi^{A_k}$. This shows that $f \leq_T A_k$ as required, and completes the proof. $\square$

Corollary 1. *For each Turing degree $\boldsymbol{a}$, let $\boldsymbol{a} = \boldsymbol{a}_0 < \boldsymbol{a}_1 < \cdots$ be the sequence of degrees defined by Proposition 1, and let $\mathcal{I}_{\boldsymbol{a}} = \{\boldsymbol{d} : (\exists n)\ \boldsymbol{d} \leq \boldsymbol{a}_n\}$ be the corresponding Scott ideal. Then $\mathcal{I}_{\boldsymbol{a}}$ is the intersection of the two lower cones*

$$\mathcal{I}_{\boldsymbol{a}} = \{\boldsymbol{d} : \boldsymbol{d} \leq \boldsymbol{b}\ \&\ \boldsymbol{d} \leq \boldsymbol{c}\},$$

with $\boldsymbol{b}$ and $\boldsymbol{c}$ defined as in Theorem 4 (hence with jumps limit-computable in $\boldsymbol{a}$, uniformly in the given $A \in \boldsymbol{a}$). $\square$

4 Applications and Questions

The first use of Scott ideals that will spring to the minds of many readers is the creation of ω-models of the axiom system $\mathbf{WKL}_0$. It is well-known that Scott ideals yield models for this system: given a Scott ideal $\mathcal{I}$, just take the model of second-order arithmetic with standard first-order part and containing those subsets of ω whose Turing degrees lie in $\mathcal{I}$. Theorem 4 offers a uniform method of producing such models.

However, the existence of such models of $\mathbf{WKL}_0$ has long been known (see [2,9], among other sources, for background), and the present author cannot see that the uniformity here adds anything significant to our understanding of those models. The motivation for the work in this article was different. The author's original purpose in establishing Theorem 4 was to use the sets B and C constructed there to define the subgroup

$$G_{BC} = \{f \in \mathrm{Aut}(\overline{\mathbb{Q}}) : f \leq_T B\ \&\ f \leq_T C\}$$

of the absolute Galois group $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ of the field $\mathbb{Q}$ of rational numbers – or equivalently, the automorphism group $\mathrm{Aut}(\overline{\mathbb{Q}})$ of the algebraic closure $\overline{\mathbb{Q}}$. Here we have fixed a computable presentation $\overline{\mathbb{Q}}$ of this algebraic closure. (In fact, $\overline{\mathbb{Q}}$ is computably categorical, so the specific choice of presentation is irrelevant.) In some respects this work follows and expands upon that in [5,6,10].

Elements of $\mathrm{Aut}(\overline{\mathbb{Q}})$, expressed as permutations of $\overline{\mathbb{Q}}$, are readily viewed as paths through a finite-branching tree T, which can be computably presented and has computable branching. The subgroup G_{BC} has the further property that, whenever a computable infinite subtree of T is computed using finitely many elements $f_1, \ldots, f_n$ of G_{BC} as parameters, that subtree will contain a path that also lies in G_{BC}. (Recall that the intersection of the lower cones below B and C has the property that, for every D in this intersection, the intersection also contains a set E having PA degree relative to D. With $D = f_1 \oplus \cdots \oplus f_n$, the corresponding E will compute the desired path.)

In forthcoming work [7], the author has shown the following.

Theorem 5. *For every Scott ideal $\mathcal{I}$ in the Turing degrees, the set*

$$Aut_I(\overline{\mathbb{Q}}) = \{f \in Aut(\overline{\mathbb{Q}}) : \deg f \in \mathcal{I}\}$$

forms a subgroup of $Aut(\overline{\mathbb{Q}})$ that is elementary for Σ_1 and Π_1 formulas and also for all positive formulas (i.e., prenex formulas in the language of fields that do not use the negation connective).

This elementarity has been extended to a further class of Σ_2 formulas, and might yet turn out to hold for more complicated formulas (allowing negation) as well. On the other hand, it is conjectured that the subgroups $\mathrm{Aut}_{\boldsymbol{d}}(\overline{\mathbb{Q}}) = \{f \in \mathrm{Aut}(\overline{\mathbb{Q}}) : \deg f \leq \boldsymbol{d}\}$ defined by principal Turing ideals may not be elementary to the same extent.

The group $\mathrm{Aut}(\overline{\mathbb{Q}})$ is naturally viewed as a profinite group: an inverse limit of finite groups, namely the Galois groups of number fields over $\mathbb{Q}$. It is hoped here that it may turn out to be productive to view $\mathrm{Aut}(\overline{\mathbb{Q}})$ simultaneously as a direct limit. The subgroups $\mathrm{Aut}_{\mathcal{I}}(\overline{\mathbb{Q}})$ of the form above, under inclusion, do form a directed system whose direct limit is $\mathrm{Aut}(\overline{\mathbb{Q}})$. So also do the subgroups $\mathrm{Aut}_{\boldsymbol{d}}(\overline{\mathbb{Q}})$, as $\boldsymbol{d}$ ranges over all Turing degrees, but using subgroups of greater elementarity appears to be a more promising path. On the other hand, for the subgroups $\mathrm{Aut}_{\boldsymbol{d}}(\overline{\mathbb{Q}})$, the directed system is well-known: it is simply the set of all degrees under Turing reducibility $\leq$, as $\mathrm{Aut}_{\boldsymbol{c}}(\overline{\mathbb{Q}}) \subseteq \mathrm{Aut}_{\boldsymbol{d}}(\overline{\mathbb{Q}})$ just if $\boldsymbol{c} \leq \boldsymbol{d}$.

For the subgroups given by Scott ideals, it is natural to use the specific Scott ideals $\mathcal{I}_{\boldsymbol{a}}$ constructed above. Clearly, under inclusion, these too form a directed system with direct limit $\mathrm{Aut}(\overline{\mathbb{Q}})$. However, the inclusion relation here seems substantially more complicated. It will be clear, first of all, that distinct degrees $\boldsymbol{a}$ and $\tilde{\boldsymbol{a}}$ may yield equal Scott ideals $\mathcal{I}_{\boldsymbol{a}} = \mathcal{I}_{\tilde{\boldsymbol{a}}}$: just run the procedure from Theorem 4 on a set $A \in \boldsymbol{a}$, and let $\tilde{A}$ be the set A_1 produced by that procedure, so that $\mathcal{I}_{\tilde{\boldsymbol{a}}}$ is defined by the increasing sequence $\boldsymbol{a}_1 < \boldsymbol{a}_2 < \cdots$.

The surprising aspect of this problem, however, is the question of whether Theorem 4 is monotonic at all.

Definition 2. *An operator F on Turing degrees (mapping each $\boldsymbol{d}$ to a Turing degree $F(\boldsymbol{d})$) is* monotonic *if*

$$(\forall \boldsymbol{c})(\forall \boldsymbol{d})\ [\boldsymbol{c} \leq \boldsymbol{d} \implies F(\boldsymbol{c}) \leq F(\boldsymbol{d})].$$

An operator G mapping Turing degrees to ideals (or other sets of degrees) is monotonic *if*

$$(\forall \boldsymbol{c})(\forall \boldsymbol{d})\ [\boldsymbol{c} \leq \boldsymbol{d} \implies G(\boldsymbol{c}) \subseteq G(\boldsymbol{d})].$$

The operator U defined using the functional Γ in the Uniform Low Basis Theorem, mapping each $\boldsymbol{a}$ to some degree **PA** relative to $\boldsymbol{a}$, is of the first type here, while the map $\boldsymbol{a} \mapsto \mathcal{I}_{\boldsymbol{a}}$ is of the second type. It is an open question (to this author's knowledge) whether either of these operators is monotonic in the sense above. The question may startle many readers, who would have assumed (as the author did at first!) that constructions such as that in the Uniform Low Basis Theorem automatically respect Turing reducibility. Recall, however, that for each $\boldsymbol{a}$, there are a wide variety of degrees **PA** relative to $\boldsymbol{a}$: indeed, even among those low relative to $\boldsymbol{a}$, there are pairs of degrees, both **PA** relative to $\boldsymbol{a}$, whose greatest lower bound under $\leq$ is $\boldsymbol{a}$ itself. Of course, when $\boldsymbol{a} \leq \tilde{\boldsymbol{a}}$, the degree $U(\tilde{\boldsymbol{a}})$, being **PA** relative to $\tilde{\boldsymbol{a}}$, computes a path through every $\boldsymbol{a}$-computable infinite subtree of $2^{<\omega}$, hence is **PA** relative to $\boldsymbol{a}$ as well – but this does not ensure that $U(\boldsymbol{a})$ will lie below $U(\tilde{\boldsymbol{a}})$. For addressing the possibility of using the groups $\mathrm{Aut}_{I_a}(\overline{\mathbb{Q}})$ to form a directed system with recognizable inclusions, therefore, it would be highly useful to have answers to the following questions about monotonicity.

Question 1. Does the Uniform Low Basis Theorem hold monotonically? That is, does there exist a Turing functional $\varPhi$ with all of the following properties?

- For every $A \subseteq \omega$, $\varPhi^A$ is total with $(\forall n)\ \lim_s \varPhi^A(n,s) = \chi_{(P_A)'}(n)$ for the jump $(P_A)'$ of some set P_A that is **PA** relative to A (as in the Uniform Low Basis Theorem); and
- When $A \leq_T B$, the sets P_A and P_B defined as above by $\varPhi$ satisfy $P_A \leq_T P_B$ (or equivalently, $(P_A)' \leq_1 (P_B)'$).

Question 2. Can we produce low Scott ideals monotonically? That is, do there exist Turing functionals $\varTheta$ and $\varGamma$ with all of the following properties?

- For every $A \subseteq \omega$, $\varTheta^A$ and $\varGamma^A$ are both total with

$$(\forall n)\ [\lim_s \varTheta^A(n,s) = \chi_{(B_A)'}(n)\ \&\ \lim_s \varGamma^A(n,s) = \chi_{(C_A)'}(n)$$

  for the jumps $(B_A)'$ and $(C_A)'$ of some sets B_A and C_A such that the set $\{\boldsymbol{d} : \boldsymbol{d} \leq \deg B_A\ \&\ \boldsymbol{d} \leq \deg C_A\}$ is a Scott ideal containing $\deg A$; and
- When $A \leq_T \widetilde{A}$, the sets B_A, C_A, $B_{\widetilde{A}}$, and $C_{\widetilde{A}}$ defined by $\varTheta$ and $\varGamma$ satisfy

$$(\forall \boldsymbol{d})\ [(\boldsymbol{d} \leq \deg B_A\ \&\ \boldsymbol{d} \leq \deg C_A) \implies (\boldsymbol{d} \leq \deg B_{\widetilde{A}}\ \&\ \boldsymbol{d} \leq \deg C_{\widetilde{A}})].$$

A positive answer to Question 1 would yield a positive answer to Question 2, by applying the procedure from Theorem 4 to the functional $\varPhi$ in Question 1.

References

1. Brattka, V., de Brecht, M., Pauly, A.: Closed choice and a uniform low basis theorem. Ann. Pure Appl. Logic **163**(8), 986–1008 (2012)
2. Hirschfeldt, D.R.: Slicing the Truth: On the Computable and Reverse Mathematics of Combinatorial Principles, Lecture Notes Series, Institute for Mathematical Sciences, National Univ. of Singapore, vol. 28. World Scientific, Singapore (2014)
3. Jockusch, C.G., Soare, R.I.: $\varPi^0_1$-classes and degrees of theories. Trans. Am. Math. Soc. **173**, 33–56 (1972)
4. Kleene, S.C., Post, E.L.: The upper semi-lattice of degrees of recursive unsolvability. Ann. Math. **59**(3), 379–407 (1954)
5. La Roche, P.: Effective Galois theory. J. Symb. Log. **46**(2), 385–392 (1981)
6. Metakides, G., Nerode, A.: Effective content of field theory. Ann. Math. Log. **17**, 289–320 (1979)
7. Miller, R.: Computability and the absolute Galois group of $\mathbb{Q}$, to appear
8. Shoenfield, J.R.: Degrees of models. J. Symb. Log. **25**, 233–237 (1960)
9. Simpson, S.G.: Subsystems of Second-Order Arithmetic, 2nd edn. Cambridge University Press, Cambridge (2010)
10. Smith, R.L.: Effective aspects of profinite groups. J. Symb. Log. **46**(4), 851–863 (1981)
11. Soare, R.I.: Recursively Enumerable Sets and Degrees. Springer, New York (1987)
12. Spector, C.: On degrees of recursive unsolvability. Ann. Math. **64**(3), 581–592 (1956)