10

Peano Arithmetic and Π^0_1-Classes

10.1 Logical Background

One of the earliest purposes of computability theory was the study of logical systems and theories. We consider theories in a computable language: one which is countable, and whose function, relation, and constant symbols and their arities are effectively given. We also assume that languages come equipped with an effective coding for formulas and sentences in the languages, i.e., a *Gödel numbering*, and identify sets of formulas with the corresponding set of Gödel numbers. We can then speak of the Turing degree of a theory in a computable language. Here we will examine the language $\mathcal{L} = \{+, \cdot, <, 0, 1\}$ of arithmetic, and theories extending PA, the theory of Peano arithmetic.

Definition 10.1.1. Let $\mathcal{D}_{\mathrm{PA}}$ be the set of (Turing) degrees of complete consistent extensions of Peano arithmetic; such a degree is called a *PA degree.*

The following is surely the best known theorem in mathematical logic.

Theorem 10.1.2 (Gödel, 1931; Rosser, 1936)**.**

1. *The theory of Peano arithmetic is incomplete.*
2. *Furthermore, any consistent computably axiomatizable extension of PA is also incomplete.*

Corollary 10.1.3. $\mathbf{0} \notin \mathcal{D}_{PA}$.

R.I. Soare, *Turing Computability*, Theory and Applications of Computability,
DOI 10.1007/978-3-642-31933-4_10

Thus, there is no complete consistent extension of PA which is computable. However, there are many ways to extend PA to a complete theory, and we can think of them as paths on a computable tree. We identify a completion of Peano Arithmetic with the set of Gödel numbers of its sentences.

10.2 Π^0_1 Classes and Completions of Theories

Theorem 10.2.1. *There exists a Π^0_1 class whose members are precisely the completions of Peano Arithmetic. Thus, $\mathcal{D}_{PA}$ is the degree spectrum of a Π^0_1 class.*

Proof. (Sketch). Fix a bijective Gödel numbering $G : \omega \to \mathrm{Sent}_{\mathcal{L}}$ for sentences of arithmetic. Given $\sigma \in 2^{<\omega}$, we identify σ with the sentence

$$\theta(\sigma) = \bigwedge_{\sigma(i)=1} G(i) \quad \& \quad \bigwedge_{\sigma(j)=0} \neg G(j).$$

We say that a sentence θ "appears to be consistent at stage t" if there is no derivation of $\neg\theta$ from the first t axioms of PA in fewer than t lines. Since there are finitely many such derivations, the relation $R(\sigma, t) =$ "$\theta(\sigma)$ appears to be consistent at stage t" is computable. Therefore, the class

$$\mathcal{C} \quad = \quad \{\, f \in 2^\omega : (\forall n)(\forall t < n) R(f \upharpoonright t, n) \,\}$$

is a Π^0_1 class. Some f is an element of this class if and only if the corresponding set of sentences $G(\{n : f(n) = 1\})$ is a complete consistent extension of PA. □

Remark 10.2.2. *This theorem follows from an analysis of Lindenbaum's Lemma. Note that no special properties of PA were used, beyond the fact that it is a computably axiomatizable theory in a computable language. Therefore, the same theorem applies to all such theories.*

Lindenbaum's Lemma says that a consistent theory T has a complete consistent extension. This follows by the Compactness Theorem.

We defined a PA degree as a degree of a completion of Peano Arithmetic. From this definition, it may be surprising that the class of degrees is closed upwards. This is true, however, and to demonstrate it we need an important fact arising from Gödel's incompleteness theorem: the proof actually constructs a "Gödel sentence" which is independent of the axioms.

Theorem 10.2.3 (Gödel's Incompleteness Theorem, effective version)**.**

From a description of a consistent, computably axiomatizable theory T extending PA, we can effectively find a sentence, called the Gödel-Rosser sentence of T*, which is independent of T.*

10.3 Equivalent Properties of PA Degrees

The PA degrees arise naturally in a variety of contexts, especially those relating to trees and weak König's lemma. This is because the PA degrees are exactly those degrees which can achieve weak König's lemma by finding paths through trees. For this reason, there are several equivalent properties which all serve to define the PA degrees. We shall highlight a few of these properties.

Definition 10.3.1. A function $f : \omega \to \omega$ is *diagonally noncomputable (d.n.c.)* if, for all e, if $\varphi_e(e)\downarrow$, then $f(e) \neq \varphi_e(e)$.

Recall that up to Turing degree this is equivalent to f being fixed point free by Exercise 5.4.5.

Definition 10.3.2. A function is *n-valued* if $f(e) < n$ for each $e \in \omega$.

The term "diagonally noncomputable" derives from the particular way that d.n.c. functions are noncomputable. We see that if f is d.n.c., f cannot be computable, because then f would be φ_e for some e, but f and φ_e differ on argument e; thus d.n.c. functions diagonalize against the list of all (partial) computable functions. We will be primarily interested in 2-valued d.n.c. functions.

Theorem 10.3.3 (Scott, 1962; Jockusch and Soare, 1972b; Solovay, unpublished)**.** [1]

For a Turing degree $\mathbf{d}$*, the following are equivalent:*

(i) $\mathbf{d}$ *is the degree of a complete consistent extension of Peano arithmetic.*

(ii) $\mathbf{d}$ *computes a complete consistent extension of Peano arithmetic.*

(iii) $\mathbf{d}$ *computes a 2-valued d.n.c. function.*

(iv) Every partial computable 2-valued function has a total $\mathbf{d}$*-computable 2-valued extension.*

(v) Every nonempty Π^0_1 *class has a member of degree at most* $\mathbf{d}$*.*

(vi) Every computably inseparable pair has a separating set of degree at most $\mathbf{d}$*.*

Proof. (i) $\implies$ (ii). This implication is trivial.

(ii) $\implies$ (iii). Let $\mathbf{d}$ compute a complete consistent extension T of PA, and let f be the (partial computable) diagonal function $f(e) = \varphi_e(e)$. By results of Gödel and Kleene, there is a formula ψ representing f, in the

[1] In 1962 Scott proved the equivalence of conditions (i) and (v). In 1972b Jockusch and Soare proved the equivalence of conditions (ii) and (vi); the equivalence with (iii) and (iv) is also implicit in their work. Jockusch and Soare left the equivalence of (i) and (ii) as an open question, which was answered by Solovay (unpublished).

sense that

$$f(x)\downarrow = y \iff PA \vdash \psi(x,y), \text{ and}$$

$$f(x)\downarrow \neq y \iff PA \vdash \neg\psi(x,y).$$

Since $PA \vdash \psi(x,y)$ implies that $\psi(x,y) \in T$, and T is complete and **d**-computable, the function

$$\widehat{f}(e) = \begin{cases} 1 & \psi(e,0) \in T \\ 0 & \neg\psi(e,0) \in T \end{cases}$$

is a **d**-computable 2-valued d.n.c. function.

(iii) $\implies$ (iv). Suppose g is a 2-valued d.n.c. function, and let f be a partial computable 2-valued function. There is a computable function $\widehat{f}$ such that $f(x) = \varphi_{\widehat{f}(x)}(\widehat{f}(x))$ for all x. Then $1 - (g \circ \widehat{f})$ is a total **d**-computable 2-valued function extending f.

(iv) $\implies$ (v). Let $\mathcal{P}$ be a nonempty Π^0_1 class, and T a computable tree with $\mathcal{P} = [T]$. Fix a computable bijection $h : \omega \to 2^{<\omega}$. Let f be the function

$$f(e) = \begin{cases} 0 & \begin{array}{l} h(e) \in T \text{ and there is a level } l \text{ such that } h(e)^\frown 0 \\ \text{has a descendent at level } l \text{ in } T, \text{ but } h(e)^\frown 1 \text{ does not} \end{array} \\ \\ 1 & \begin{array}{l} h(e) \in T \text{ and there is a level } l \text{ such that } h(e)^\frown 1 \\ \text{has a descendent at level } l \text{ in } T, \text{ but } h(e)^\frown 0 \text{ does not.} \end{array} \end{cases}$$

This function f is partial computable, since to compute $f(e)$ one simply searches for a level l such that one case or the other holds. If $h(e) \in T$ is extendible, then either both $h(e)^\frown 0$ and $h(e)^\frown 1$ are extendible, in which case $f(e)\uparrow$, or only one is, so $f(e)\downarrow$, and $h(e)^\frown f(e)$ is extendible. Let $\widehat{f}$ be a 2-valued **d**-computable extension of f. Then using $\widehat{f}$, we can find an element of $[T]$ as follows: starting with any string $\sigma \in T^{\text{ext}}$, apply $\widehat{f} \circ h^{-1}$ to get either 0 or 1, which we can append to σ to get a longer string still in T^{ext}. Starting with the empty string, we can iterate this process to get an infinite **d**-computable path through $[T]$, i.e., an element of $\mathcal{P}$.

(v) $\implies$ (vi). If A, B is a computably inseparable pair, the class of separating sets is a Π^0_1 class by Theorem 9.3.2. If property (v) holds, this has a **d**-computable member.

(vi) $\implies$ (i). Fix some order of $\mathcal{L}$-sentences, and some order for generating proofs. Let A be the set of pairs (F, ψ), where F is a finite set of $\mathcal{L}$-sentences and ψ is an $\mathcal{L}$-sentence, such that a proof of a contradiction is found from $\text{PA} \cup F \cup \{\psi\}$ before (if ever) finding a proof of a contradiction from $\text{PA} \cup F \cup \{\neg\psi\}$. Similarly, let B be the set of pairs (F, ψ), such that a proof of contradiction is found from $\mathrm{PA} \cup F \cup \{\neg\psi\}$ before (if ever) finding one from $\mathrm{PA} \cup F \cup \{\psi\}$. Clearly A and B are disjoint c.e. sets. Suppose the pair A, B has a $\mathbf{d}$-computable separating set C. Let $D \in \mathbf{d}$. We shall construct a completion T of PA, of degree $\mathbf{d}$, in stages, along with a bijective function $g : \omega \to \mathrm{Sent}_{\mathcal{L}}$, also defined in stages. At stage n we shall determine $g(n)$, and decide whether $g(n) \in T$. Define the set of sentences,

$$F_n \quad = \quad (T \cap g[0 \dots n-1]) \ \cup \ \{\neg\psi : \psi \in g[0 \dots n-1] \ \setminus T\}.$$

In other words, F_n keeps track of every sentence we decided by the beginning of stage n. It contains those sentences we have declared to be in T, together with the negations of those sentences we have declared not to be in T. At stage n, do the following:

1. If n is even, let $g(n)$ be the Gödel sentence of $\mathrm{PA} \cup F_n$. If n is odd, let $g(n)$ be the first $\mathcal{L}$-sentence not yet in the range of g.

2. If $n = 2s$ is even, consider whether s is an element of D. If $s \in D$, then $g(n) \in T$; otherwise, $g(n) \notin T$.

3. If n is odd, consider the pair $(F_n, g(n))$. If this pair is in C, then $g(n) \notin T$; otherwise, $g(n) \in T$.

We shall show that T is a complete consistent extension of PA, of degree $\mathbf{d}$. Assume (for the sake of induction) that F_n is consistent with PA. (Since $F_0 = \emptyset$, it is consistent with PA.) Note that F_{n+1} is either $F_n \cup \{g(n)\}$ or else $F_n \cup \{\neg g(n)\}$. Since F_n is consistent with PA, at least one of $F_n \cup \{g(n)\}$ and $F_n \cup \{\neg g(n)\}$ must be consistent with PA. Furthermore, if n is even, both are consistent since $g(n)$ is the Gödel sentence for $\mathrm{PA} \cup F_n$. If both are consistent with PA, then clearly F_{n+1} is as well. Suppose instead only one of the two is consistent (so we know n is odd). If only $F_n \cup \{g(n)\}$ is consistent with PA, then a proof of contradiction will be found from $PA \cup F_n \cup \{\neg g(n)\}$ before finding one from $PA \cup F_n \cup \{g(n)\}$, so $(F_n, g(n)) \in B$. Thus $(F_n, g(n)) \notin C$; by the construction, $g(n) \in T$, and F_{n+1} is consistent with PA. Similarly, if only $F_n \wedge \neg g(n)$ is consistent with PA, then the construction goes the opposite way and again F_{n+1} is consistent with PA. By induction, F_n is consistent with PA for all n, so $T = \bigcup_n F_n$ is consistent with PA. Since F_n decides $g(0) \dots g(n-1)$, T is complete. Therefore, T is a complete consistent extension of PA.

In order to show that T has degree $\mathbf{d}$, we first show that $g \leq_T T$. To see this, note that $g(n)$ is either the first $\mathcal{L}$-sentence which is not one of $g(0) \dots g(n-1)$, if n is odd, or else $g(n)$ is the Gödel sentence of $\mathrm{PA} \cup F_n$, where F_n is determined entirely by T and the values $g(0) \dots g(n-1)$. Thus $g(n)$ can be computed from n, $g(0) \dots g(n-1)$, and T, so $g \leq_T T$. From the construction, we see that $s \in D$ if and only if $g(2s) \in T$, so we have $D \leq_T g \oplus T \leq_T T$. However, the entire construction was $\mathbf{d}$-computable, so $T \in \mathbf{d}$. $\square$