(3.1)

Standard Representations

In mathematical logic, a Gödel numbering is a function that assigns each symbol and well-formed formula to a unique natural number, used by K. Gödel for the proof of his incompleteness theorems in 1931.

Equivalently, a Gödel numbering $\varphi : \mathbb{N} \to P^{(1)}$ of the set $P^{(1)}$ of partial recursive functions (computable number functions) $f :\subseteq \mathbb{N} \to \mathbb{N}$ is defined uniquely up to equivalence in Type-1 recursion theory. This is due to the utm-theorem and smn-theorem.

The utm-theorem (Universal Turing Machine Theorem) affirms the existence of a computable universal function which is capable of calculating any other computable function, an abstraction of the Universal Turing Machine.

In practical terms, the smt-theorem (Translation Lemma or Parameter Theorem) says that for a given programming language and $m, n \in \mathbb{N}$, there exists a particular algorithm that accepts as input the source code of a program with $m+n$ free variables together with m values. This algorithm generates source code that substitutes the values for the first m free variables, leaving the rest free.

Notation and Representation

In order to produce Type-2 generalizations of these theorems, we must first introduce general notations and representations. A "notation" of a set M is a surjective function $\nu :\subseteq \Sigma_1^{*} \to M$, and a "representation" is a surjective function $\delta :\subseteq \Sigma_1^{\omega} \to M$. A "naming system" is a notation or a representation.

Sometimes we will say that $p \in Y$ is a γ-name of $x \in M$ if $\gamma :\subseteq Y \to M$ is a naming system and $\gamma(p) = x$. Notice that we do not consider functions $\gamma :\subseteq \Sigma_1^{*} \cup \Sigma_1^{\omega} \to M$ as names.

For $\gamma :\subseteq Y \to M$, $\gamma' :\subseteq Y' \to M'$ with $Y, Y' \in \{\Sigma_1^{*}, \Sigma_1^{\omega}\}$, we define:

(1) $f :\subseteq Y \to Y'$ reduces γ to γ' iff $\gamma(y) = (\gamma' f)(y)$ for all $y \in dom(\gamma)$.

We write $\gamma \leq \gamma'$ if f is computable and $\gamma \leq_t \gamma'$ if f is continuous. $\leq$ and $\leq_t$ are preorders on the class of naming systems.

(2) $\gamma \equiv \gamma'$ iff $\gamma \leq \gamma' \leq \gamma$, and $\gamma \equiv_t \gamma'$ iff $\gamma \leq_t \gamma' \leq_t \gamma$. $\equiv$ and $\equiv_t$ are equivalences on the class of naming systems.

(3-3)

utm/smn - Properties Revisited

In elementary computability theory, Turing machines computing word functions $f :\subseteq \Sigma_1^{*} \to \Sigma_1^{*}$ are encoded canonically by words from Σ_1^{*}. If $\psi_w :\subseteq \Sigma_1^{*} \to \Sigma_1^{*}$ is the word function computed by the Turing machine with code w, then the partial function (partial notation) $w \mapsto \psi_w$ of the computable word functions satisfies the utm and smn-properties:

utm(ψ): The function $(w, x) \mapsto \psi_w(x)$ is computable.

smn(ψ): For every computable function $f :\subseteq \Sigma_1^{*} \times \Sigma_1^{*} \to \Sigma_1^{*}$, there is a total computable function $r : \Sigma_1^{*} \to \Sigma_1^{*}$ with $r(x) \in \mathrm{dom}(\psi)$ for all $x \in \Sigma_1^{*}$ and $f(x,y) = \psi_{r(x)}(y)$ for all $x, y \in \Sigma_1^{*}$.

Thm utm(ψ) and smn(ψ) hold.

Proof: See any standard text for detail, including for detail of universal encoding. $\square$

(3.4)

General utm/smn-Properties

Let $a,b,c \in \{*,\omega\}$, G^{ab} be a set of functions $g :\subseteq \Sigma_1^{a} \to \Sigma_1^{b}$, $\zeta :\subseteq \Sigma_1^{c} \to G^{ab}$ be a naming system of G^{ab}. Define:

utm(ζ): There is a computable (universal) function $u :\subseteq \Sigma_1^{c} \times \Sigma_1^{a} \to \Sigma_1^{b}$ with $\zeta_x(y) = u(x,y)$ for all $x \in \mathrm{dom}(\zeta)$ and $y \in \Sigma_1^{a}$.

smn(ζ): For every computable function $f :\subseteq \Sigma_1^{c} \times \Sigma_1^{a} \to \Sigma_1^{b}$ there is a total computable function $s : \Sigma_1^{c} \to \Sigma_1^{c}$ such that $s(x) \in \mathrm{dom}(\zeta)$ for all $x \in \Sigma_1^{c}$ and $f(x,y) = \zeta_{s(x)}(y)$ for all $x \in \Sigma_1^{c}$ and $y \in \Sigma_1^{a}$.

Consider a canonical encoding of Type-2 machines with one input tape by words $w \in \Sigma_1^{*}$ such that the set TC of code words is recursive. For all $a,b \in \{*,\omega\}$, define:

$$P^{ab} := \{ f :\subseteq \Sigma_1^{a} \to \Sigma_1^{b} \mid f \text{ is computable} \}$$

and a notation $\xi^{ab} : \Sigma_1^{*} \to P^{ab}$ of the set P^{ab} defined by $\xi^{ab}(w)$ is undefined on non-code words, otherwise the function $f \in P^{ab}$ computed by the Type-2 machine with code w.

<u>Thm</u> For all $a,b \in \{*,\omega\}$, we have utm(ξ^{ab}) and smn(ξ^{ab}).

Classes of Continuous Functions

As well as representations of functions, we also would like representations of sets of functions. Obviously, we can only consider sets of size at most the continuum cardinality, as we have discussed.

Define:

$$F^{**} := \{f \mid f :\subseteq \Sigma_1^{*} \to \Sigma_1^{*}\},$$
$$F^{*\omega} := \{f \mid f :\subseteq \Sigma_1^{*} \to \Sigma_1^{\omega}\},$$
$$F^{\omega *} := \{f \mid f :\subseteq \Sigma_1^{\omega} \to \Sigma_1^{*} \text{ continuous} \wedge \mathrm{dom}(f) \text{ open}\},$$
$$F^{\omega\omega} := \{f \mid f :\subseteq \Sigma_1^{\omega} \to \Sigma_1^{\omega} \text{ continuous} \wedge \mathrm{dom}(f)\ G_\delta\}.$$

Thm $F^{\omega *} = \{h_* \mid h :\subseteq \Sigma_1^{*} \to \Sigma_1^{*} \text{ has prefix-free domain}\}$
$F^{\omega\omega} = \{h_\omega \mid h : \Sigma_1^{*} \to \Sigma_1^{*} \text{ is monotone}\}$

By the next theorem, $F^{\omega *}$, $F^{\omega\omega}$ represent essentially all partial and continuous functions $f :\subseteq \Sigma_1^{\omega} \to \Sigma_1^{*}$ and $f :\subseteq \Sigma_1^{\omega} \to \Sigma_1^{\omega}$, respectively.

Thm Every ctns $f :\subseteq \Sigma_1^{\omega} \to \Sigma_1^{*}$ has an extension in $F^{\omega *}$ and every ctns $f :\subseteq \Sigma_1^{\omega} \to \Sigma_1^{\omega}$ has an extension in $F^{\omega\omega}$.

Finally, functions from F^{ab} are essentially closed under composition for $a, b \in \{*, \omega\}$.

Thm If $g \in F^{ab}$, $f \in F^{bc}$ where $a, b, c \in \{*, \omega\}$. If $b = \omega$ and $c = *$ then $f \circ g$ has an extension $d \in F^{a*}$ with $\mathrm{dom}(d) \cap \mathrm{dom}(g) = \mathrm{dom}(f \circ g)$. Otherwise, $f \circ g \in F^{ac}$.

(3.6)

Standard Representations of F^{ab}

For all $a, b \in \{*, \omega\}$, define the standard rep. $\eta^{ab}: \Sigma_1^{\omega} \to F^{ab}$ of F^{ab} by:

$$\eta^{ab}(\langle x, p \rangle)(y) := \xi_x^{\omega b}\langle p, y \rangle$$

for all $x \in \Sigma_1^{*}$, $p \in \Sigma_1^{\omega}$, $y \in \Sigma_1^{a}$, and

$$\eta^{ab}(q)(\_) := \bot$$

if for no $x \in \Sigma_1^{*}$, $\iota(x)$ is a prefix of q.

Therefore, roughly speaking, $\eta^{ab}_{\langle x, q \rangle}(y)$ is the result of the Type-2 machine M with code x applied to input.

Lem Representations η^{ab} are well-defined.

Lem Computable functions have computable names. That is, a function $f :\subseteq \Sigma_1^{a} \to \Sigma_1^{b}$ is computable iff $f = \eta^{ab}_p$ for some computable $p \in \Sigma_1^{\omega}$.

Thm For each $a, b \in \{*, \omega\}$, we have $utm(\eta^{ab})$ and $smn(\eta^{ab})$.

(3-7)

An Equivalence Theorem

In 1967, H. Rogers proved that for the numbering $\varphi : \mathbb{N} \to P^{(1)}$, up to equivalence, is the only numbering of $P^{(1)}$ satisfying the utm-theorem and smn-theorem. This can be easily generalised to our naming systems of function spaces.

Thm For notations β, γ, δ of G^{ab} with $utm(\delta)$ and $smn(\delta)$, we have:

$$\begin{aligned}
&(1) \quad (\beta \leq \gamma \wedge utm(\gamma)) \Rightarrow utm(\beta), \\
&(2) \quad (smn(\beta) \wedge \beta \leq \gamma) \Rightarrow smn(\gamma), \\
&(3) \quad (utm(\beta) \wedge smn(\gamma)) \Rightarrow \beta \leq \gamma, \\
&(4) \quad utm(\beta) \Leftrightarrow \beta \leq \delta, \\
&(5) \quad smn(\beta) \Leftrightarrow \delta \leq \beta, \\
&(6) \quad (utm(\gamma) \wedge smn(\gamma)) \Leftrightarrow \gamma \equiv \delta.
\end{aligned}$$

Finally, we can think about this theorem and the last, purely topologically.

Thm We have $tutm(\eta^{ab})$ and $tutm(\eta^{ab})$, where, for a representation of F^{ab}, δ:

$tutm(\delta)$: There is a $u \in F^{\omega b}$ s.t. $u\langle p, y\rangle = \delta_p(y)$ for all $p \in dom(\delta)$ and $y \in \Sigma_1^{a}$.

$tsmn(\delta)$: For every $f \in F^{\omega b}$, there is a ctns $S : \Sigma_1^{\omega} \to \Sigma_1^{\omega}$ with $S(p) \in dom(\delta)$ and $f\langle p, y\rangle = \delta_{S(p)}(y)$ for all $p \in \Sigma_1^{\omega}$ and $y \in \Sigma_1^{a}$.

3.8

Decidable Sets

Recall from Type-1 theory that a subset $A \subseteq \Sigma_1^*$ is called recursive or decidable iff its characteristic function is computable, and is called recursively enumerable (r.e.) iff it is the domain of a computable function $f :\subseteq \Sigma_1^* \to \Sigma_1^*$. A is recursive iff A and A^c are both r.e.

We will generalise these concepts for Type-2 theory, opting for the terminology of "decidable", ~~Reg..~~ defining "recursive" later.

Consider $X \subseteq Z \subseteq Y := Y_1 \times \cdots \times Y_k$ where $k \geq 1$ and $Y_1, \ldots, Y_k \in \{\Sigma_1^*, \Sigma_1^\omega\}$. Then:

(1) X is called r.e. open in Z iff $X = dom(f) \cap Z$ for some computable function $f :\subseteq Y \to \Sigma_1^*$.

(2) X is called decidable in Z iff both X and $Z \setminus X$ are r.e. open in Z.

(3) X is open in Z iff $X = U \cap Z$ for some open set $U \subseteq Y$, and closed in Z iff $Z \setminus X$ is open in Z.

Observe that X r.e. open (decidable) in Z $\Rightarrow$ X is open (clopen) in Z. This follows from our theorem from 2.15.

3-9

A Characterization Theorem

Thm Let $X \subseteq Z \subseteq Y$ be as before. Then X is clopen (decidable) in Z iff there is a continuous (computable) function $f :\subseteq Y \to \Sigma_1^*$ with $f(z) = 1$ if $z \in X$ and $f(z) = 0$ if $z \in Z \setminus X$.

$\Rightarrow$ Proof 1: Let X and $Z \setminus X$ be open in Z. Then there are open sets U, V such that $X = U \cap Z$ and $Z \setminus X = V \cap Z$. Define $f :\subseteq Y \to \Sigma_1^*$ by $dom(f) := Z$, $f(z) := 1$ if $z \in X$, $f(z) := 0$ if $z \in Z \setminus X$. Then clearly f meets the requirements and is continuous since $f^{-1}[\{1\}] = U \cap dom(f)$, $f^{-1}[\{0\}] = V \cap dom(f)$.

$\Leftarrow$ Let f be as above. Then $f^{-1}[\{1\}] = U \cap dom(f)$ for some open set U. So $X = f^{-1}[\{1\}] \cap Z = U \cap Z$, thus X is open in Z. Similarly, $Z \setminus X$ is open in Z. $\square$

$\Rightarrow$ Proof 2: Let X and $Z \setminus X$ be r.e. open in Z. Then there are Type-2 machines M and N with $dom(f_M) \cap Z = X$ and $dom(f_N) \cap Z = Z \setminus X$. Clearly we can construct a machine L that runs M and N in parallel, halting as soon as either halts, writing either 1 or 0 if M or N halts first, respectively. Set $f := f_L$.

$\Leftarrow$ Let L be the machine that computes f, as above. From L, we construct M which on input $z \in Y$ halts iff L halts on z with result 1. Then $X = dom(f_M) \cap Z$, that is, X is r.e. open in Z. Similarly, $Z \setminus X$ is r.e. open in Z. $\square$

Some Examples

Recall that each decidable set $X \subseteq \Sigma_1^\omega$ has the form $X = A\Sigma_1^\omega$ for some finite $A \subseteq \Sigma_1^*$.

Eg 1 Clearly Z is decidable in Z.

Eg 2 If X is r.e. open (decidable), then X is r.e. open (decidable) in Z whenever $X \subseteq Z$.

Eg 3 The set $X = \{p \in \Sigma_1^\omega \mid p \neq 0^\omega\}$ is r.e. open, but its complement is not, since it does not contain a set $w\Sigma_1^\omega$. Therefore, X is not closed, so not clopen, so not decidable.

Eg 4 $X \subseteq Y_1 \times \cdots \times Y_k$ is r.e. open (decidable) iff $\langle X \rangle$ is r.e. open (decidable) in $\langle Y_1 \times \cdots \times Y_k \rangle$.

Eg 5 Union and intersection of r.e. open sets are r.e. open. For decidable sets, complement is decidable too.

Eg 6 If $f :\subseteq Y_1 \times \cdots \times Y_k \to Y_0$ is computable and $U \subseteq W \subseteq Y_0$ is r.e. open (decidable) in W, then $f^{-1}[U]$ is r.e. open (decidable) in $f^{-1}[W]$.

(3.11)

Another Characterization Theorem

Thm For $X \subseteq Y := Y_1 \times \cdots \times Y_k$ ($Y_1, \ldots, Y_k \in \{\Sigma_1^{*}, \Sigma_1^{\omega}\}$), the following properties are equivalent:

(1) X is r.e. open
(2) $X = A \circ Y$ for some r.e. $A \subseteq (\Sigma_1^{*})^k$
(3) X is open and $\{y \in (\Sigma_1^{*})^k \mid y \circ Y \subseteq X\}$ is r.e.

Recursive Open Sets

For any open $X \subseteq Y := Y_1 \times \cdots \times Y_k$, define:

X is r.e. $:\Leftrightarrow \{y \in (\Sigma_1^{*})^k \mid y \circ Y \subseteq X\}$ is r.e.

X is co-r.e. $:\Leftrightarrow \{y \in (\Sigma_1^{*})^k \mid y \circ Y \not\subseteq X\}$ is r.e.

X is recursive $:\Leftrightarrow \{y \in (\Sigma_1^{*})^k \mid y \circ Y \subseteq X\}$ is decidable.

Thus, an open set $X \subseteq Y$ is recursive iff it is both r.e. and co-r.e. Notice that if Y is $(\Sigma_1^{*})^k$, then X is r.e. iff it is r.e. in the usual sense, and similarly for recursive.

Recall that the r.e. open subsets of Y are closed under union and intersection. The recursive open sets are not closed under union, but are under intersection. Decidable $\Rightarrow$ Recursive, but the reverse is not necessarily true.

3.12

Taking Stock

So far, we have defined computability on the sets Σ_1^* of finite words and Σ_1^ω of infinite sequences by Type-2 machines. In TTE we induce computability on other sets M by using finite or infinite words as "names". Machines, therefore, still transform "concrete" sequences of symbols. We have seen that some concepts of computability theory have similar topological counterparts.

In particular, we are interested in computability concepts induced by naming systems. Our next steps must be to transfer our notions for Σ_1^* and Σ_1^ω to sets M by means of naming systems. We will then introduce and discuss the important class of "admissible" representations, at which point we will then be able to return to our original motivating questions.

For means of simplicity, we do not consider naming systems where both finite and infinite symbol sequences are used as names simultaneously. If $\gamma :\subseteq Y \to M$ is a naming system, then every element of M is named since γ is surjective, but not every $y \in Y$ is a name, necessarily. Furthermore, an element of M may have several names. We have informally seen this for the reals, represented by decreasing rational intervals.

Summary of Concepts

From Type-1 theory, we have Turing Machines, partial (computable) functions, numberings, and recursive and r.e. sets $A \subseteq \Sigma_1^*$.

From Topology, we have continuity and compactness.

From Type-2 theory, we have:

(1) Type-2 Machines
(2) (Computable) String Functions
(3) (Computable) Elements
(4) Tupling Functions
(5) Essentially closure under composition
(6) Primative recursion
(7) The finiteness property
(8) Discrete and Cantor topologies
(9) Computable $\Rightarrow$ Continuous
(10) Naming and reduction
(11) UTM and smn-theorems
(12) Classes of (continuous) functions
(13) (Continuous) extension
(14) r.e. open and decidable
(15) co-r.e. open and recursive open

(3.14)

Appendix: Proof of Theorem

For simpliciity we consider only case $Y = \Sigma_1^{*} \times \Sigma_1^{\omega}$.

$1 \Rightarrow 2$ Let X be r.e. open. Then $X = dom(f_M)$ for some M. Define a computable function $g :\subseteq \Sigma_1^{*} \times \Sigma_1^{*} \to \Sigma_1^{*}$ by:

$$g(x_1, x_2) := \begin{cases} 1 & \text{if } M \text{ halts on input } (x_1, x_2 0^\omega) \text{ after reading at most the first } |x_2| \text{ symbols from the second input tape} \\ \bot & \text{otherwise} \end{cases}$$

$A := g^{-1}[\{1\}]$ is r.e.. If $(x_1, x_2) \in A$, then M halts on input $(x_1, x_2 q)$ for every $q \in \Sigma_1^{\omega}$, so $(x_1, x_2) \circ Y \subseteq X$, thus $A \circ Y \subseteq X$. On the other hand, if $(x_1, p) \in X$, then for some prefix x_2 of p, M halts on input (x_1, p) after reading at most the first $|x_2|$ symbols from the second input tape. Therefore, $(x_1, x_2) \in A$ and $(x, p) \in A \circ Y$, thus $X \subseteq A \circ Y$. □

$2 \Rightarrow 1$ Let M be a TM with $A = dom(f_M)$, and N the Type-2 machine which on input $(x, p) \in \Sigma_1^{*} \times \Sigma_1^{\omega}$ searches for the smallest $n = \langle k, m \rangle$ such that M on $(x, p_{<k})$ halts in m steps as soon as it has found n. So $A \circ Y = dom(f_N)$. □

$2 \Rightarrow 3$ Suppose $y \circ Y \subseteq A \circ Y$. Since $y \circ Y$ is a compact subset of Y, there is a finite C s.t. $y \circ Y \subseteq C \circ Y$. Since this property is decidable in y and C, if A is r.e. then $\{y \in (\Sigma_1^{*})^k \mid y \circ Y \subseteq A \circ Y\}$ is. Clearly X is open if $X = A \circ Y$. □

$3 \Rightarrow 2$ If X is open and $B := \{y \in (\Sigma_1^{*})^k \mid y \circ Y \subseteq X\}$, then $X = B \circ Y$. □