Recursion Theory

Recap

Martin Zimmermann

Universität des Saarlandes

July 30th, 2015

Computability, Decidability, Enumerability

Without formalizing the notion of "algorithm" yet, we say that..

- .. an algorithm $\mathcal{A}$ computes a partial function $f\colon \mathbb{N}^n \to \mathbb{N}$, if $\mathcal{A}$ terminates on input $\overline{x}$ iff $f(\overline{x})$ is defined. Furthermore, if $f(\overline{x})$ is defined, then $\mathcal{A}$ has to return $f(\overline{x})$. We say f is *computable*.
- .. an algorithm $\mathcal{A}$ decides a set $A \subseteq \mathbb{N}^n$, if $\mathcal{A}$ on input $\overline{x}$ terminates and returns 1 if $\overline{x} \in A$ and 0 if $\overline{x} \notin A$. We say A is decidable.
- .. an algorithm $\mathcal{A}$ enumerates a set $A \subseteq \mathbb{N}^n$, if $\mathcal{A}$ started without input returns the members of A one after the after (in any order). We say that A is enumerable.

1. A decidable iff A and $\mathbb{N}^n \setminus A$ enumerable.
2. A enumerable iff $A = \mathrm{dom}(f)$ for some computable f.

Unlimited Register Machines

Very simple notion of algorithm (no distinction between machine and code!)

An unlimited register machine (URM) is a list $\mathcal{P} = I_1, \ldots, I_k$ of instructions such that $I_k = k$ `STOP` and for every $j < k$ is

- $I_j = j$ $\texttt{INC(X}_\texttt{i}\texttt{)}$,
- $I_j = j$ $\texttt{DEC(X}_\texttt{i}\texttt{)}$, or
- $I_j = j$ $\texttt{IF X}_\texttt{i}\texttt{>0 GOTO}$ ℓ

with $\ell \in \{1, \ldots, k\}$ and $i \geq 1$.

Configuration: $(k_0, k_1, k_2, \ldots)$ where

- $k_0 \in \{1, \ldots, k\}$ is the current line number, and
- $k_i \in \mathbb{N}$ is the current content of register i.

URM - Semantics

Successor configuration of $(k_0, k_1, k_2, \ldots)$:

- none, if $k_0 = k$ (termination)
- $(k_0 + 1, k_1, \ldots, k_{i-1}, k_i + 1, k_{i+1}, \ldots)$, if $I_j = j$ `INC(X_i)`
- $(k_0 + 1, k_1, \ldots, k_{i-1}, k_i \dot{-} 1, k_{i+1}, \ldots)$, if $I_j = j$ `DEC(X_i)`
- $(k_0', k_1, k_2, \ldots)$, if $I_j = j$ `IF X_i>0 GOTO` ℓ, where $k_0' = \ell$, if $k_i > 0$, and $k_0' = k_0 + 1$, if $k_i = 0$.

- $\mathcal{P} x_1 \cdots x_n \downarrow$, if $\mathcal{P}$ starting with conf. $(1, x_1, \ldots, x_n, 0, \ldots)$ reaches configuration of the form $(k, y_1, y_2, \ldots)$.
- Then: $\mathcal{P} x_1 \cdots x_n \downarrow y_1$.
- Otherwise: $\mathcal{P} x_1 \cdots x_n \uparrow$.

Encoding URM's

- The pairing function $[x, y] = 2^x(2y + 1) - 1$ is bijective.
- The inverses
  - $[z]_1 =$ "exponent of 2 in prime factorization of $z + 1$" and
  - $[z]_2 = (z + 1)/[z]_1$

  satisfy $[[z]_1, [z]_2] = z$.

- The sequence encoding $\langle \varepsilon \rangle = 0$ and

  $$\langle x_0, \ldots, x_k \rangle = p_0^{x_0} \cdots p_{k-1}^{x_{k-1}} p_k^{x_k+1} - 1$$

  is bijective.

Encoding URM's cont'd

Instructions encoded by

- $\text{code}(\texttt{INC(X}_\texttt{i}\texttt{)}) = [0, i]$
- $\text{code}(\texttt{DEC(X}_\texttt{i}\texttt{)}) = [i, 0]$
- $\text{code}(\texttt{IF X}_\texttt{i}\texttt{>0 GOTO } \ell) = [i, \ell]$
- $\text{code}(\texttt{STOP}) = 0$

URM $\mathcal{P} = I_1, \ldots, I_k$ encoded by

$$\text{code}(\mathcal{P}) = \langle \text{code}(I_1), \ldots, \text{code}(I_k) \rangle$$

- Some e do not encode an URM.
- Let $\mathcal{P}_e$ be URM encoded by e, if e encodes one, otherwise an always non-terminating URM.
- $\varphi_e^{(n)}$: n-ary function computed by $\mathcal{P}_e$, e is an index of $\varphi_e^{(n)}$.
- $W_e^{(n)}$: domain of $\varphi_e^{(n)}$.

Primitive Recursive Functions

Base functions:

- Successor $S(x) = x + 1$
- n-ary constant zero $c_0^{(n)}(\overline{x}) = 0$
- i-th n-ary projection $p_i^{(n)}(x_1, \ldots, x_n) = x_i$ for $1 \leq i \leq n$

Composition: given $h\colon \mathbb{N}^m \to \mathbb{N}$ and $g_1, \ldots, g_m\colon \mathbb{N}^n \to \mathbb{N}$ define $f\colon \mathbb{N}^n \to \mathbb{N}$ via

$$f(\overline{x}) = h(g_1(\overline{x}), \ldots, g_m(\overline{x}))$$

Primitive recursion: given $g\colon \mathbb{N}^n \to \mathbb{N}$ and $h\colon \mathbb{N}^{n+2} \to \mathbb{N}$ define $f\colon \mathbb{N}^{n+1} \to \mathbb{N}$ via

$$f(\overline{x}, 0) = g(\overline{x})$$

$$f(\overline{x}, y + 1) = h(\overline{x}, y, f(\overline{x}, y))$$

Primitive Recursive Functions

Definition

1. A function is primitive recursive, if it can be constructed from the base functions, composition, and primitive recursion.
2. A set $A \subseteq \mathbb{N}^n$ is primitive recursive, if its characteristic function is primitiverecursive.

Syntactic Sugar:

- Bounded sum and product, e.g., $\mathrm{sum}_g(\overline{x}, y) = \sum_{z=0}^{y} g(\overline{x}, y, z)$
- Bounded quantification, e.g., $Q_E\overline{x}y \Leftrightarrow \exists z \leq y \; R\overline{x}yz$
- Bounded minimization, e.g.,

$$\mu z \leq y \; R\overline{x}yz = \min\left(\{z \leq y \mid R\overline{x}yz\} \cup \{y\}\right)$$

- Case distinction (if the conditions are primitive recursive)

μ-recursive Functions

Unbounded minimization (μ-operator): given $g\colon \mathbb{N}^{n+1} \to \mathbb{N}$ define $f\colon \mathbb{N}^n \to \mathbb{N}$ via

$$f(\overline{x}) = \min\{y \mid g(\overline{x}, y) = 0 \text{ and } g(\overline{x}, z) > 0 \text{ for all } z < y\}$$

where $\min \emptyset = \bot$.

Notation: $f(\overline{x}) = \mu y\colon g(\overline{x}, y) = 0$

Definition

1. A function is μ-recursive, if it can be constructed from the base functions, composition, primitive recursion, and unbounded minimization.
2. A set $A \subseteq \mathbb{N}^n$ is μ-recursive, if its characteristic function is μ-recursive.

Kleene Normal Form Theorem

Theorem

There is a primitive recursive function $U\colon \mathbb{N} \to \mathbb{N}$ *and a primitive recursive predicate* $T_n \subseteq \mathbb{N}^{n+2}$ *for every* n *such that for every* n*-ary URM-computable function* f*, we have*

$$f(\overline{x}) = U(\mu y\ T_n e \overline{x} y)$$

for some suitable e *(an index of* f*).*

Note: U and T_n independent on f, only e depends on f.

- Converse is true as well: every μ-recursive function is URM-computable.
- Corollary: every μ-recursive function is definable with just one application of the μ-operator.

Non-recursive Sets

- $K = \{e \mid \varphi_e(e)\downarrow\} = \{e \mid e \in W_e\} = \{e \mid \exists y T_1 eey\}$.
- $\mathrm{Tot} = \{e \mid \varphi_e \text{ total}\} = \{e \mid W_e = \mathbb{N}\} = \{e \mid \forall x \exists y T_1 exy\}$.

Theorem

1. *K is not recursive.*
2. Tot *is not recursive.*

Universal Functions

Let $\mathcal{R}$ be a class of n-ary functions. $F\colon \mathbb{N}^{n+1} \to \mathbb{N}$ is universal for $\mathcal{R}$, if

- for every $f \in \mathcal{R}$ there exists e such that $f(\overline{x}) = F(e, \overline{x})$, and
- for every e, the function $\overline{x} \mapsto F(e, \overline{x})$ is in $\mathcal{R}$.

Let $\Phi^{(n+1)}\colon \mathbb{N}^{n+1} \to \mathbb{N}$ be defined by $\Phi(e, \overline{x}) = U(\mu y T_n e \overline{x} y)$.

Theorem

Φ^{n+1} *is universal for the class of n-ary μ-recursive functions, and is itself μ-recursive.*

Theorem

1. *There is no universal function for the class of n-ary total μ-recursive functions that is itself total and μ-recursive.*
2. *There is no universal function for the class of n-ary p.r. functions that is itself total and p.r..*

s-m-n, Rice and Recursion Theorem

Theorem

For all $m, n \geq 1$ there is a total μ-recursive $s_n^m \colon \mathbb{N}^{m+1} \to \mathbb{N}$ such that $\varphi_e^{(m+n)}(\overline{x}, \overline{y}) = \varphi_{s_n^m(e,\overline{x})}^{(n)}(\overline{y})$ for all $\overline{x} \in \mathbb{N}^m, \overline{y} \in \mathbb{N}^n$.

- $A \subseteq \mathbb{N}$ represents semantic program property, if there is a class $\mathcal{F}$ of functions s.t. $e \in A \Leftrightarrow \varphi_e \in \mathcal{F}$.
- A is non-trivial, if $A \neq \emptyset$ and $A \neq \mathbb{N}$.

Theorem

Every non-trivial set that represents a semantic program property is not μ-recursive.

Theorem

For every total μ-recursive $f : \mathbb{N} \to \mathbb{N}$ there is an e s.t. $\varphi_e = \varphi_{f(e)}$.

Enumerable Sets

Theorem

Let $A \subseteq \mathbb{N}$ be non-empty. The following three conditions are equivalent:

1. *There is a total μ-recursive $f : \mathbb{N} \to \mathbb{N}$ s.t. $A = f(\mathbb{N})$.*
2. *There is a μ-recursive $f : \mathbb{N} \to \mathbb{N}$ s.t. $A = f(\mathbb{N})$.*
3. *There is a μ-recursive $f : \mathbb{N} \to \mathbb{N}$ s.t. $A = dom(f)$.*

Such an A is called (μ-recursively) enumerable.

Theorem

K is enumerable.

Reductions

Let $A, B \subseteq \mathbb{N}$.

- $A \leq_m B$ if there exists a total μ-recursive $f\colon \mathbb{N} \to \mathbb{N}$ such that $x \in A \Leftrightarrow f(x) \in B$.
- $A \leq_1 B$ if there exists an injective total μ-recursive $f\colon \mathbb{N} \to \mathbb{N}$ such that $x \in A \Leftrightarrow f(x) \in B$.

Lemma

1. *$\leq_1$, $\leq_m$ are reflexive and transitive.*
2. *$A \leq_1 B$ implies $A \leq_m B$.*
3. *$A \leq_1 B$ iff $\overline{A} \leq_1 \overline{B}$.*
4. *$A \leq_m B$ iff $\overline{A} \leq_m \overline{B}$.*
5. *$A \leq_m B$ and B μ-recursive, then A μ-recursive.*
6. *$A \leq_m B$ and B enumerable, then A enumerable.*

Completeness

Theorem
$A, \overline{A}$ enumerable iff A μ-recursive.

Hence: $\overline{K}$ not enumerable.

A is 1-complete (m-complete) if

- A is enumerable, and
- $B \leq_1 A$ ($B \leq_m A$) for all enumerable B.

Theorem
K is 1-complete and m-complete.

Degrees and Myhill's Theorem

- $A \equiv_1 B$ iff $A \leq_1 B$ and $B \leq_1 A$.
- $A \equiv_m B$ iff $A \leq_m B$ and $B \leq_m A$.

Equivalence classes are called degrees.

A and B are recursively isomorphic, if there is a recursive bijection $f : \mathbb{N} \to \mathbb{N}$ such that $x \in A \Leftrightarrow f(x) \in B$.

Theorem

$A \equiv_1 B$ iff A and B are recursively isomorphic.

Productive and Creative Sets

- $A \subseteq \mathbb{N}$ is productive, if there is a recursive $f : \mathbb{N} \to \mathbb{N}$ such that

$$W_x \subseteq A \Rightarrow f(x)\downarrow \text{ and } f(x) \in A \setminus W_x.$$

- $A \subseteq \mathbb{N}$ is creative, if A is enumerable and $\overline{A}$ is productive.

Lemma

Productive sets are not enumerable.

Theorem

1. *K is creative.*
2. *If $A \leq_m B$ and A productive, then B productive.*

Immune and Simple Sets

Theorem

1. *Every infinite enumerable set contains an infinite recursive subset.*
2. *Every productive set contains an infinite enumerable subset.*

- $A \subseteq \mathbb{N}$ is immune, if it is infinite and contains no infinite enumerable subset.
- $A \subseteq \mathbb{N}$ is simple, if it is enumerable and $\overline{A}$ is immune.

Theorem

There exists a simple set S.

Comparing Degree Structures

Theorem
There are sets $A \subseteq \mathbb{N}$ and $B \subseteq \mathbb{N}$ such that $A \equiv_m B$ and $A \not\equiv_1 B$.

Theorem
$A \subseteq \mathbb{N}$ is creative iff A is m-complete.

Theorem
Every simple set is enumerable, not recursive, but also not m-complete.

Truth-table Reductions

- A truth-table condition is a pair $((y_1, \ldots, y_n), \alpha)$ where $\alpha\colon \mathbb{B}^n \to \mathbb{B}$.
- $B \subseteq \mathbb{N}$ satisfies the condition, if $\alpha(\chi_B(y_1), \ldots, \chi_B(y_n)) = 1$.
- tt-conditions are enumerable.

$A \leq_{tt} B$ if there exists a total recursive function $f\colon \mathbb{N} \to \mathbb{N}$ such that $x \in A \Leftrightarrow B$ satisfies tt-condition $f(x)$.

Lemma

1. *$\leq_{tt}$ is reflexive and transitive.*
2. *$A \leq_m B$ implies $A \leq_{tt} B$ but the converse is not necessarily true.*
3. *$A \leq_{tt} \overline{A}$ for every $A \subseteq \mathbb{N}$.*
4. *$A \leq_{tt} B$ and B μ-recursive, then A μ-recursive.*

Turing Reductions

Consider URMs with additional oracle instructions

j `IF X_i` $\in B$ `GOTO` ℓ

- $\mathcal{P}_e^B$: program with code e and oracle B
- φ_e^B: unary function computed by $\mathcal{P}_e^B$.
- W_e^B: domain of φ_e^B.

1. $A \leq_T B$, if $\chi_A = \varphi_e^B$ for some e (A recursive in B).
2. A enumerable in B, if $A = W_e^B$ for some e.

Friedberg & Muchnik

Lemma

1. *$A, \overline{A}$ enumerable in A iff A recursive in B.*
2. *$\leq_T$ is reflexive and transitive.*
3. *$A \leq_{tt} B$ implies $A \leq_T B$, but the converse is not necessarily true.*
4. *$A \leq_T B$ and B μ-recursive, then A μ-recursive.*

- $A \equiv_T B$ if $A \leq_T B$ and $B \leq_T A$.
- Equivalence classes are Turing degrees.
- A Turing degree is enumerable, if it contains an enumerable set.

Theorem

There are incomparable enumerable Turing degrees.

The Turing Jump

For $A \subseteq \mathbb{N}$ let $A' = \{x \in \mathbb{N} \mid \varphi_x^A(x)\downarrow\}$ and $A^{(n)} = A^{\overbrace{\prime\cdots\prime}^{n \text{ times}}}$.

Lemma

1. $A \leq_T A'$, *but* $A' \not\leq_T A$.
2. $\emptyset' \equiv_T K$.

Theorem

Let B be enumerable in A. Then, $B \leq_m A'$.

The Arithmetic Hierarchy

Define hierarchy Σ^0_n, Π^0_n of relations via

- $\Sigma^0_0 = \Pi^0_0$ are the class of recursive relations.
- Σ^0_{n+1} is the class of relations R such that

  $$R\overline{x} \Leftrightarrow \exists y_1 \cdots \exists y_m Q\overline{x}y_1 \dots y_m$$

  for some $Q \in \Pi^0_n$.
- Π^0_{n+1} dually with universal quantifiers and $Q \in \Sigma^0_n$.
- Define $\Delta^0_n = \Sigma^0_n \cap \Pi^0_n$.

R is arithmetic if it is in some Σ^0_n.

The Arithmetic Hierarchy

Lemma

1. $R \in \Sigma_n^0 \Leftrightarrow \overline{R} \in \Pi_n^0$
2. $\Sigma_n^0, \Pi_n^0, \Delta_n^0$ *are closed under union and intersection.*
3. $\Sigma_n^0, \Pi_n^0 \subsetneq \Delta_{n+1}^0 \subsetneq \Sigma_{n+1}^0, \Pi_{n+1}^0$.
4. $A \in \Sigma_n^0$ *implies* $T^A \in \Delta_{n+1}^0$.
5. $\emptyset^{(n)} \in \Sigma_n^0$.

Theorem

1. $A \in \Sigma_{n+1}^0$ *iff* A *is enumerable in some* $B \in \Sigma_n^0$.
2. $A \in \Delta_{n+1}^0$ *iff* $A \leq_T B$ *for some* $B \in \Sigma_n^0$.
3. $A \in \Sigma_n^0$ *iff* $A \leq_m \emptyset^{(n)}$.
4. $A \in \Delta_{n+1}^0$ *iff* $A \leq_T \emptyset^{(n)}$.