Lessons touched by this meeting according to schedule:

- 12. 25/11/2024
  - Exercises
- 13. 26/11/2024
  - Recursive sets. Reduction. [§7.1, see also §6.1 and §9.1]

DEFINITION 13.1. A set $A \subseteq \mathbb{N}$ is *recursive* if its characteristic function

$$\chi_A : \mathbb{N} \to \mathbb{N}$$

$$\chi_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

is computable.

In other words, a set is recursive if there exists an algorithm (computable function) that can decide membership in the set - given any x ∈ ℕ, it can determine in a finite number of steps whether x belongs to A or not.

The notion of recursiveness has several important implications:

1. Decidability: The membership problem "x ∈ A?" for a recursive set A is decidable. An algorithm exists that always terminates and correctly answers yes or no.
2. Closure properties: The class of recursive sets is closed under complement, union and intersection. If A and B are recursive, then so are $\bar{A}$, A ∪ B and A ∩ B.
3. Simple sets: All finite sets and some easily describable infinite sets like ℕ itself are recursive. The set of prime numbers is also recursive.

On the other hand, the following sets are not recursive:

(a) $K = \{x \mid x \in W_x\}$, since

$$\chi_K(x) = \begin{cases} 1 & x \in W_x \\ 0 & x \notin W_x \end{cases}$$

is not computable;

(b) $\{x \mid \varphi_x \text{ total}\}$

Another important implication:

Reductions: If A ≤_m B (A many-one reduces to B) and B is recursive, then A is also recursive. Conversely, if A is not recursive and A ≤_m B, then B is not recursive either. This allows proving non-recursiveness of sets.

DEFINITION 13.5. Let $A, B \subseteq \mathbb{N}$. We say that the problem $x \in A$ *reduces* to the problem $x \in B$ (or simply that A reduces to B), written $A \leqslant_m B$ if there exists a function $f : \mathbb{N} \to \mathbb{N}$ computable and total such that, for every $x \in \mathbb{N}$

$$x \in A \quad \Leftrightarrow \quad f(x) \in B$$

In this case, we say that f is the *reduction function*.

Consider an example from the lesson:

EXAMPLE 13.7. $K \leqslant_m T = \{x \mid \varphi_x \text{ total}\}$

PROOF. We prove that there exists $s : \mathbb{N} \to \mathbb{N}$ computable and total such that $x \in k \Leftrightarrow s(x) \in T$. In other words

$$x \in W_x \Leftrightarrow \varphi_{f(x)} \text{ is total}$$

To do so, we can define

$$g(x, y) = \begin{cases} 1 & x \in W_x \\ \uparrow & \text{otherwise} \end{cases}$$

which is computable, since

$$g(x, y) = \mathbf{1}(\varphi_x(x)) = \mathbf{1}(\Psi_U(x, x))$$

Then, by the *smn*-theorem we have that there exists $s : \mathbb{N} \to \mathbb{N}$ computable and total such that

$$\varphi_{s(x)}(y) = g(x, y)$$

and

$$x \in K \Rightarrow x \in W_x \Rightarrow \forall y\ \varphi_{s(x)}(y) = g(x, y) = 1 \Rightarrow \varphi_{s(x)} \text{ total } \Rightarrow s(x) \in T$$
$$x \notin K \Rightarrow x \notin W_x \Rightarrow \forall y\ \varphi_{s(x)}(y) = g(x, y) \uparrow \Rightarrow \varphi_{s(x)} \text{ not total } \Rightarrow s(x) \notin T$$

□

Let's jump immediately to related exercises:

Exercise 7.12. Prove that a set $A \subseteq \mathbb{N}$ is recursive if and only if $A \leqslant_m \{0\}$.

To prove that a set A ⊆ ℕ is recursive if and only if A ≤_m {0}, we will show both implications.

(⇒) Assume A is recursive. Then its characteristic function χ_A is computable. Define the reduction function f : ℕ → ℕ as f(x) = 1 - χ_A(x). Clearly, f is computable (composition of computable functions). Now, x ∈ A ⟺ χ_A(x) = 1 ⟺ f(x) = 0 ⟺ f(x) ∈ {0}. Thus, A ≤_m {0} via f.

(⇐) Assume A ≤_m {0} via a computable function f. Then x ∈ A ⟺ f(x) ∈ {0} ⟺ f(x) = 0. So we can write χ_A(x) = sg(f(x)), which is computable. Hence, A is recursive.

Therefore, A ⊆ ℕ is recursive if and only if A ≤_m {0}.

Exercise 7.13. Let $A \subseteq \mathbb{N}$ be a non-empty set. Prove that A is recursively enumerable if and only if there exists a function $f : \mathbb{N} \to \mathbb{N}$ such that $dom(f)$ is the set of prime numbers and $img(f) = A$.

Let ∅ ≠ A ⊆ ℕ and suppose A is recursively enumerable. We want to prove that there exists a function f : ℕ → ℕ such that dom(f) is the set of prime numbers and img(f) = A.

Since A is r.e., there exists e ∈ ℕ such that A = W_e = dom(ϕ_e). Let p_i denote the i-th prime number. Define f as:

f(x) = ϕ_e(μi. x = p_i)

In other words, f(x) = ϕ_e(i) where i is the index such that x is the i-th prime number.

We claim that f satisfies the required conditions:

1. dom(f) = {p_i | i ∈ ℕ}, the set of all prime numbers, by definition of f.
2. img(f) = A.
    - If y ∈ A, then y = ϕ_e(i) for some i. Let x = p_i. Then f(x) = f(p_i) = ϕ_e(i) = y. So y ∈ img(f).
    - If y ∈ img(f), then y = f(x) for some prime x = p_i. Thus, y = f(p_i) = ϕ_e(i) ∈ A.

Conversely, if A is not r.e., such an f cannot exist, because img(f) is always r.e. (direct image of r.e. set under computable function is r.e.). Thus, the claim holds.

From one last year exam:

c. Show that given $A, B \subseteq \mathbb{N}$, if A is recursive and $B = A \cap \mathbb{P}$ then B is recursive (here $\mathbb{P}$ denotes the set of even numbers). Does the converse hold? I.e., is it the case that if $B = A \cap \mathbb{P}$ is recursive then A is recursive?

Let $A, B \subseteq \mathbb{N}$ with A recursive and let $B = A \cap \mathbb{P}$. Just observe that $\mathbb{P}$ is recursive (in fact $\chi_{\mathbb{P}}(y) = \overline{sg}(rm(2, y))$). Hence B is the intersection of recursive sets which is known to be recursive.

The converse is false. In fact, consider the set $A = \{2x + 1 \mid x \in K\}$. We have that $B = A \cap \mathbb{P} = \varnothing$ is recursive. However, A is not recursive since $K \leqslant_m A$. The reduction function can simply be $f(x) = 2x + 1$. Clearly it is total and computable and $x \in K$ iff $f(x) \in A$.

Let's jump to other *exercises* of various kinds:

EXERCISE (ASSIGNED): Is it the case that every computable function is the composition of two non-computable functions?

Proof:

Let f : ℕ → ℕ be a computable function. We define the functions g and h as follows:

g(x) = f(x) if x ∉ K

0 if x ∈ K

h(x) = x (identity function)

Here, K is the halting set, i.e., K = {x ∈ ℕ | ϕ_x(x) ↓}.

First, let's verify that f = g ∘ h:

(g ∘ h)(x) = g(h(x))

= g(x)

= f(x) if x ∉ K

0 if x ∈ K

= f(x) for all x, since f is total

Now, we prove that g and h are not computable.

h is clearly computable, as it is the identity function.

Assume, for the sake of contradiction, that g is computable. Then we could use g to decide the halting set K as follows:

For any x ∈ ℕ, compute g(x) and f(x).

If g(x) = f(x), then x ∉ K.

If g(x) = 0 and f(x) ≠ 0, then x ∈ K.

This procedure would give us a computable characteristic function for K:

χ_K(x) = 1 if g(x) = 0 and f(x) ≠ 0

0 if g(x) = f(x)

However, we know that K is not recursive (as proven using the recursion theorem in a previous example). Therefore, our assumption that g is computable must be false.

Thus, g is not computable, and f is the composition of the non-computable function g and the computable function h.

Since f was arbitrary, this proves that every computable function is the composition of two non-computable functions.

Exercise 2

State the s-m-n theorem and use it to prove that there exists a total computable function $s : \mathbb{N} \to \mathbb{N}$ such that $W_{s(x)} = \mathbb{P}$ and $E_{s(x)} = \{z \in \mathbb{N} \mid z \geqslant x\}$ (where again $\mathbb{P}$ is the set of even numbers).

Solution: We can define, for instance,

$$f(x, y) = \begin{cases} x + y/2 & \text{if } y \in \mathbb{P} \\ \uparrow & \text{otherwise} \end{cases}$$

which is clearly computable. In fact

$$f(x, y) = x + qt(2, y) + \mu w.rm(2, y)$$

Seen as a function of y, it has as domain $\mathbb{P}$ and as codomain $\{z \mid z \geqslant x\}$. Then one can use the smn theorem to get a function $s : \mathbb{N} \to \mathbb{N}$ such that for all $x, y \in \mathbb{N}$

$$\varphi_{s(x)}(y) = f(x, y) = \begin{cases} x + y/2 & \text{if } y \in \mathbb{P} \\ \uparrow & \text{otherwise} \end{cases}$$

Then s is the desired function, in fact:

- $W_{s(x)} = \mathbb{P}$, by construction;
- $E_{s(x)} = \{\varphi_{s(x)}(y) \mid y \in W_{s(x)}\} = \{\varphi_{s(x)}(y) \mid y \in \mathbb{P}\} = \{x + y/2 \mid y \in \mathbb{P}\} = \{x + y' \mid y' \in \mathbb{N}\} = \{z \mid z \geqslant x\}$, as desired.

The key relationship between decidability/semi-decidability and recursive/recursively enumerable sets can be expressed through characteristic functions and predicates (important for later reasoning)

A set $A \subseteq N$ is recursive (decidable) if and only if its characteristic function χ_A is computable:

A set $A \subseteq N$ is recursively enumerable (r.e.) or semi-decidable if and only if its semi-characteristic function sc_A is computable:

sc_A(x) = {

1 if $x \in A$

↑ if $x \notin A$

}

The Structure Theorem for semi-decidable predicates states that $P(\vec{x})$ is semi-decidable if and only if there exists a decidable predicate $Q(t,\vec{x})$ such that:

$P(\vec{x}) \equiv \exists t.Q(t,\vec{x})$

This is crucial because it:

1. Characterizes semi-decidable predicates in terms of decidable ones via existential quantification

2. Shows that semi-decidable predicates can be expressed as projections of decidable predicates

3. Leads to the Projection Theorem which states that if $P(x,\vec{y})$ is semi-decidable, then $\exists x.P(x,\vec{y})$ is also semi-decidable

These theorems provide powerful tools for:

- Proving predicates are semi-decidable by expressing them in terms of decidable predicates

- Showing closure properties of semi-decidable predicates under existential quantification

- Understanding the relationship between decidability and semi-decidability

- Constructing new semi-decidable predicates from existing ones

The theorems also help explain why semi-decidable predicates are not closed under complementation and universal quantification, which is key for understanding undecidability results.

Example of usage of such notions:

Exercise (30-06-2020)

Given two functions $f, g: \mathbb{N} \to \mathbb{N}$ with f total, define predicate $Q_{f_g}(x) = "f(x) = g(x)"$. Show that if f and g are computable, then Q_{f_g} is semidecidable. Does the converse hold, so if $Q_{f\,g}$ is semidecidable, can we deduce f and g are computable?

Solution

Let f, g be computable functions. Let $e_1, e_2 \in \mathbb{N}$ $s.t. f = \phi_{e_1}$ and $g = \phi_{e_2}$.

Then $sc_{f_g} = \mathbf{1}(\mu w. |f(x) - g(x)|$ is computable, hence Q_{f_g} is semidecidable.

If Q_{f_g} is semidecidable and let e be an index of semicharacteristic function of Q, namely $\phi_e = sc_{Q_{f_g}}$

We have $f(x) = (\mu w. H(e, x, (w)_1, (w)_2) \vee H(e, y, (w)_1, (w)_3)$ which shows f and g are computable.

Coming back to other stuff:

Exercise 6.29. Is there a total non-computable function $f : \mathbb{N} \to \mathbb{N}$ such that the function $g : \mathbb{N} \to \mathbb{N}$ defined, for each $x \in \mathbb{N}$, by $g(x) = f(x) \dot{-} x$ is computable? Provide an example or prove that such a function does not exist.

Solution: Consider $f(x) = \chi_K(x)$. Then $f(x) \dot{-} x$ is the constant 0 for each $x \geqslant 1$, therefore computable. □

Exercise 6.31. Is there a computable function $f : \mathbb{N} \to \mathbb{N}$ such that $dom(f) = K$ and $cod(f) = \mathbb{N}$? Justify your answer.

Solution: Yes, it exists. For example, consider $f(x) = \varphi_x(x)$. Clearly $dom(f) = K$. Furthermore, for each $k \in N$, if we consider an index of the constant function k we have that $f(e) = \varphi_e(e) = k$. Thus $cod(f) = \mathbb{N}$.

Alternatively one can define

$$f(x) = (\mu t. H(x, x, t)) - 1$$

Clearly$dom(f) = K$ since $f(x)\downarrow$ if there exists some t such that $H(x, x, t)$, i.e., if $x \in K$. Furthermore, for each $x \in \mathbb{N}$ just take the program Z_k which consists of $Z(1)$ repeated x times. For the corresponding index $y = \gamma(Z_k)$ we will have $f(y) = k - 1$, which shows that $cod(f) = \mathbb{N}$. □

Exercise 3.2(p). State the theorem s-m-n and use it to prove that it exists a total computable function $s : \mathbb{N} \to \mathbb{N}$ such that $|W_{s(x)}| = 2x$ and $|E_{s(x)}| = x$.

This one is also present inside 2019-09-17 exam.

Given $m, n \geq 1$ there is a total computable function $s_{m,n}: \mathbb{N}^{m+1} \to \mathbb{N}$ such that $\forall \vec{x} \in \mathbb{N}^m, \forall \vec{y} \in \mathbb{N}^n, \forall e \in \mathbb{N}$

$$\phi_e^{(m+n)}(\vec{x}, \vec{y}) = \phi_{s_{m,n}(e,\vec{x})}^{(n)}(\overrightarrow{y})$$

Given the domain should be $2x$, we find a function in which we can parametrize a value $< 2x$; given the range is x, it's simply a function which allows us to be defined computably over x. Let's give

$$g(x, y) = \begin{cases} qt(x, y), & y < 2x \\ \uparrow, & otherwise \end{cases}$$

$g(x, y)$ is computable and $sg(y) * qt(x, y) + \mu z.(y + 1 \dot{-} 2x)$ is computable itself, hence giving as range x.

By the smn-theorem, there is a computable function $g: \mathbb{N} \to \mathbb{N}\ s.t.\ \phi_{s(x)}(y) = g(x, y)\ \forall x, y \in \mathbb{N}$. Therefore, for each function:

- $W_x = \{y \mid (g(x, y) \downarrow\} = \{y \mid y < 2x\}$
- $E_{k(n)} = \{g(x, y) \mid x \in W_{s(x)}\} = \{qt(2, y) \mid y < 2x\} = \{y + 1 \dot{-} 2x \mid y + 1 < 2x\} = [0, 2x)$

as desired.

Exercise (2018-11-20-parziale)

State the smn-theorem. Use it for proving there exists $k: \mathbb{N} \to \mathbb{N}$ total and computable s.t. $\forall n \in \mathbb{N}$ we have $W_{k(n)} = \{z^n \mid z \in \mathbb{N}\}$ and $E_{k(n)}$ is the set of odd numbers

Solution

The smn-theorem states that, given $m, n \geq 1$ there is a computable total function $s_{m,n}: \mathbb{N}^{m+1} \to \mathbb{N}$ $s.t. \forall e \in \mathbb{N}, \vec{x} \in \mathbb{N}^m, \vec{y} \in \mathbb{N}^n$

$$\phi_e^{m+n}(\vec{x}, \vec{y}) = \phi_{s_{m,n}(e,\vec{x})}^{(n)}(\vec{y})$$

Define a two-arguments total-computable function $f(n, x)$ respecting the conditions:

$$f(n, x) = \begin{cases} 2z + 1, & if\ x = z^n\ for\ some\ z \\ \uparrow, & otherwise \end{cases} = 2 * (\mu z. |x \dot{-} z^n|) + 1$$

By the smn-theorem, there exists a total and computable function $k: \mathbb{N} \to \mathbb{N}$ s.t. $\phi_{k(n)}(x) = f(n, x)\ \forall n, x \in \mathbb{N}$. So, as desired:

- $W_{k(n)} = \{x \mid f(n, x) \downarrow\} = \{x \mid \exists z \in \mathbb{N}. x = z^n\} = \{z^n \mid z \in \mathbb{N}\}$
- $E_{k(n)} = \{f(n, x) \mid x \in W_{k(n)}\} = \{2\sqrt[n]{z^n} + 1 \mid z \in \mathbb{N}\} = \{2z + 1 \mid z \in \mathbb{N}\}$

Exercise (15-07-2020)

Let $A, B \subseteq \mathbb{N}$. Define reduction for $A \leq_m B$. Is it true that if A is recursive and B is finite, not empty then $A \leq_m B$? Show it or give a counterexample. And without finiteness hypothesis for B? In case in general it doesn't hold with B infinite and give a condition which allows to restore the property

Let's tackle this exercise step by step, as requested.

1. Define reduction for A ≤_m B:

We say that A ≤_m B (A is many-one reducible to B) if there exists a computable function f : ℕ → ℕ such that for all x ∈ ℕ, x ∈ A ⟺ f(x) ∈ B.

2. Is it true that if A is recursive and B is finite, not empty then A ≤_m B?

No, this is not true in general. Here's a counterexample:

Let A = ℕ\{0} (the natural numbers without 0) and B = {0}.

A is recursive (its characteristic function is computable) and B is finite, not empty.

But $A \not\leq_m B$. If there were a reduction $f : \mathbb{N} \to \mathbb{N}$ with $x \in A \iff f(x) \in B$, then $f(x)$ would have to be 0 for all $x \in A$. But $0 \notin A$, so there can't be such an f.

3. Without finiteness hypothesis for B:

If we don't assume B to be finite, then the statement "if A is recursive and B is not empty then $A \leq_m B$" does hold.

Here's why:

Let b be any element of B (which exists as B is not empty). Define $f : \mathbb{N} \to \mathbb{N}$ as follows:

$f(x) = b$ if $x \in A$

some fixed element not in B if $x \notin A$

Since A is recursive, f is computable. And $x \in A \iff f(x) = b \in B$. So f is a reduction from A to B.

4. Condition which allows to restore the property when B infinite:

The property "if A is recursive then $A \leq_m B$" holds whenever B is not empty and $A \neq \mathbb{N}\setminus\{0\}$.

Indeed, if $A = \emptyset$, the constant function $f(x) = b \in B$ is a reduction.

If $A \neq \emptyset$ and $A \neq \mathbb{N}\setminus\{0\}$, choose $a \in A$ and $a' \notin A$. Then the function

$f(x) = b$ if $x \in A$

b' if $x \notin A$

where $b \in B$ and $b' \notin B$ is a computable reduction from A to B.

So in summary, the reduction property fails only in the specific case where $A = \mathbb{N}\setminus\{0\}$ and B is a singleton. In all other cases where A is recursive and B is not empty, we can find a reduction from A to B.