Introduction to computability
Tutorial 8

Recursive functions and Uncomputability

20 November 2018

1. Show that the following functions are primitive recursive:
   a) $square(x)$ that is equal to 1 if x is a square and 0 otherwise;
   b) $sumSquares(x) = \sum_{i=0}^{x} i^2$;
   c) $mod(x, y)$ that computes the remainder of the division of x by y;
   d) $divides(x, y)$ that is equal to 1 if x divides y and 0 otherwise;
   e) bounded maximization:

$$\nu i \leqslant m\, q(\bar{n}, i) = \begin{cases} \text{the greatest } i \leqslant m \text{ such that } q(\bar{n}, i) = 1 \\ 0 \text{ if there is no such } i \end{cases} ;$$

   f) $gcd(x, y)$ that computes the greatest common divisor of x and y;
   g) $lcm(x, y)$ that computes the least common multiple of x and y.

2. Let $f : \mathbb{N} \to \mathbb{N}$ be a function such that there exists $p \in \mathbb{N}_0$ such that

$$(\forall x \in \mathbb{N})(f(x+p) = f(x)).$$

Show that f is a primitive recursive function.

3. Let Σ be an alphabet with k symbols and let $f(x, y)$ be a function where

- x is the Gödel number of a word w on the alphabet Σ
- $y \in \mathbb{N}$

The function $f(x, y)$ is the Gödel number of the word encoded by x without the y last symbols. Formally, if $w = w_0 \dots w_l$, then $f(x, y)$ is the Gödel number of the word

$$w' = \begin{cases} w_0 \dots w_{l-y} & \text{if } y \leqslant l \\ \varepsilon & \text{if } y > l \end{cases}.$$

Is this a primitive recursive function?

Gödel number

Alphabet Σ of size k. Each symbol of Σ is represented by an integer between 1 and k. The representation of a string $w = w_0 \ldots w_l$ is thus

$$gd(w) = \sum_{i=0}^{l} (k+1)^{l-i} gd(w_i).$$

3. Let Σ be an alphabet with k symbols and let $f(x, y)$ be a function where

- x is the Gödel number of a word w on the alphabet Σ
- $y \in \mathbb{N}$

The function $f(x, y)$ is the Gödel number of the word encoded by x without the y last symbols. Formally, if $w = w_0 \dots w_l$, then $f(x, y)$ is the Gödel number of the word

$$w' = \begin{cases} w_0 \dots w_{l-y} & \text{if } y \leqslant l \\ \varepsilon & \text{if } y > l \end{cases}.$$

Is this a primitive recursive function?

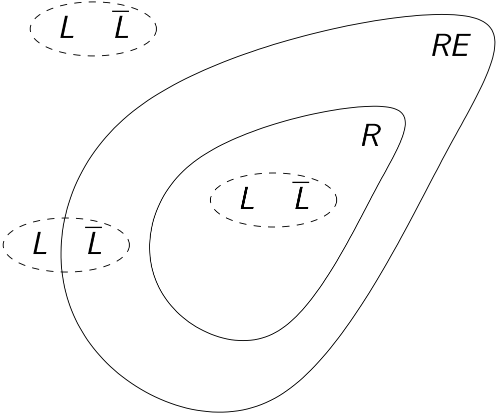
Accepted and decided languages (informally)

- A TM **decides** a language L if:
  1. it accepts all the words of the language,
  2. it rejects all other words, and
  3. it has no infinite executions.

  $\hookrightarrow$ Decidability class R **decidable** (**recursive**).
- A TM **accepts** a language L if:
  1. it accepts all the words of the language,
  2. it rejects or does not terminate for all other words.

  $\hookrightarrow$ Decidability class RE **partially decidable** (**recursively enumerable**).
- $\boxed{R \subset RE.}$

Three possibilities



Uncomputability

4. Show that the regular languages and the context-free languages are decidable.

Uncomputability

4. Show that the regular languages and the context-free languages are decidable.

5. Let L_1 and L_2 be two languages that belong to the class RE. Show that the languages $L_1 \cup L_2$ and $L_1 \cap L_2$ also belong to the class RE.

Uncomputability

4. Show that the regular languages and the context-free languages are decidable.

5. Let L_1 and L_2 be two languages that belong to the class RE. Show that the languages $L_1 \cup L_2$ and $L_1 \cap L_2$ also belong to the class RE.

6. Let M_1 and M_2 be two Turing machines.
Show that the problem whether or not there exists a word w such that M_1 and M_2 both terminate on w is undecidable.

Reduction technique

To prove that a language L_2 is undecidable ($\notin R$) knowing that L_1 is undecidable:

1. By contradiction, assume that L_2 is decidable;
2. Write an algorithm that decides L_1 using as a sub-program an algorithm that decides L_2;
3. However, L_1 is undecidable $\rightarrow$ contradiction !
   $\rightarrow$ L_2 is undecidable.

Some undecidable languages

$$\textit{universal language: } UL = \{\langle M, w\rangle \mid M \text{ accepts } w\}$$

$$\overline{UL} = \{\langle M, w\rangle \mid M \text{ rejects or cycles on } w\}$$

$$\textit{halting problem: } H = \{\langle M, w\rangle \mid M \text{ stops on } w\}$$

$$\textit{empty-word halting problem: } \{M \mid M \text{ stops on } \varepsilon\}$$

$$\textit{existential halting problem: } \{M \mid (\exists w)\ M \text{ stops on } w\}$$

$$\textit{universal halting problem: } \{M \mid (\forall w)\ M \text{ stops on } w\}$$

$$\textit{empty accepted language: } \{M \mid L(M) = \emptyset\}$$

$$\textit{recursive accepted language: } \{M \mid L(M) \in R\}$$

$$\textit{undecidable accepted language: } \{M \mid L(M) \notin R\}$$

Uncomputability

4. Show that the regular languages and the context-free languages are decidable.

5. Let L_1 and L_2 be two languages that belong to the class RE. Show that the languages $L_1 \cup L_2$ and $L_1 \cap L_2$ also belong to the class RE.

6. Let M_1 and M_2 be two Turing machines.
Show that the problem whether or not there exists a word w such that M_1 and M_2 both terminate on w is undecidable.

Bonus Exercise 9

Let M_1 and M_2 be two Turing machines that accept respectively the languages L_1 and L_2.
Prove that the problem of determining if $L_1 \cap L_2 = \emptyset$ is undecidable.