Lecture 12 – Undecidable problems, Post's correspondence problem

NTIN071 Automata and Grammars

Jakub Bulín (KTIML MFF UK)

Spring 2024

** Adapted from the Czech-lecture slides by Marta Vomlelová with gratitude. The translation, some modifications, and all errors are mine.*

Recap of Lecture 11

- Recursively enumerable languages are exactly those generated by (Type 0) grammars
  - TM to G: simulate moves on a reversed non-terminal copy of ω, generate sufficient space, cleanup if accepting state
  - G to TM: generate all strings, check if any of them represents a valid derivation of ω (sentential forms separated by $\#$)
- Context-sensitive languages:
  - context-sensitive grammars are equivalent to monotone grammars
  - Linear Bounded Automaton (LBA): nondeterministic TM with tape limited to the length of input
  - constructions: monotone grammar to LBA, LBA to monotone grammar
- Intro to computability: an overview
- decision problem $\leftrightsquigarrow$ the language of all 'YES' instances
- machine-readable encoding of TMs

The Diagonal language

Let $\mathrm{decode}(w)$ be the TM M such that $\mathrm{code}(M) = w$. (Recall: if w is not a valid code, then $\mathrm{decode}(w)$ is a fixed one-state TM with no instructions.) Then:

$$L_D = \{w \mid w \notin L(\mathrm{decode}(w))\}$$

Theorem

L_D is not recursively enumerable.

Proof idea: there cannot exist a TM recognizing L_D: running it on its own code would lead to Barber's paradox

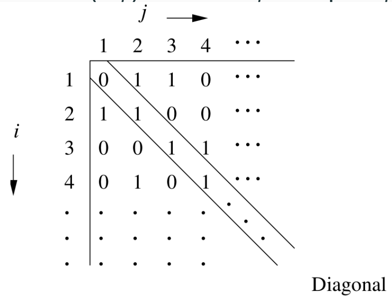
> *"The program accepts all programs that don't accept themselves. Does the program accept itself?"*

Proof that $L_D = \{w \mid w \notin L(\mathrm{decode}(w))\}$ is not RE

Proof: Assume for contradiction that $L_D = L(M)$ for some M. Let $w = \mathrm{code}(M)$. Then $L_D = \{w \mid w \notin L(M)\}$. Is $w \in L(M)$?

$$w \in L(M) \;\Leftrightarrow\; w \in L_D \;\Leftrightarrow\; w \notin L(M) \qquad \square$$

Why 'diagonal'? A variant of Cantor's diagonal argument. Order all TMs by $M_i = \mathrm{decode}(w_i)$. Does M_i accept w_i?



A TM for L_D would be one of the rows but differs from each row in the diagonal element. (Same as the proof that $\mathbb{R}$ is uncountable.)

The Universal Turing Machine

Summary of Lecture 12

- the Diagonal language L_D is not recursively enumerable
- the Universal language L_U, the Universal TM: simulate any M on any w
- Post's theorem: L recursive iff both $L, \overline{L}$ are RE
- L_U, $\overline{L_D}$ are recursively enumerable but not recursive