CSCI 3130: Formal Languages and Automata Theory — Assignment 5
The Chinese University of Hong Kong, Fall 2022 — due 11:59pm Saturday November 26

Collaborating on homework is encouraged, but you must write your own solutions in your own words and list your collaborators. Copying someone else's solution will be considered plagiarism and may result in failing the whole course.

Please answer clearly and concisely. Explain your answers. Unexplained answers will get lower scores or even no credits.

(1) (30 points) Prove that the following languages are decidable, by giving algorithms to decide them.

(a) $L_1 = \{\langle N\rangle \mid N \text{ accepts at least one string containing the substring } \texttt{bb}\}$
Here N is an NFA over the alphabet $\{\texttt{a}, \texttt{b}\}$.

(b) $L_2 = \{\langle D\rangle \mid D \text{ is a DFA that accepts only a finite number of strings}\}$

(c) $L_3 = \{\langle G, k\rangle \mid G \text{ is a CFG that generates all strings of length } k\}$

(2) (40 points) For each of the following languages, say whether it is decidable. Justify your answer in about 5–10 sentences.

(a) $L_1 = \{\langle M\rangle \mid \text{Turing machine } M \text{ accepts all palindromes (and perhaps others)}\}$
Recall that a string w is a palindrome if it reads the same forward and backward.

(b) $L_2 = \{\langle M\rangle \mid \text{Turing machine } M \text{ accepts an infinite number of strings}\}$

(c) $L_3 = \{\langle G_1, G_2\rangle \mid G_1, G_2 \text{ are CFG's such that } L(G_1) \subseteq L(G_2)\}$

(d) $L_4 = \{\langle M, t\rangle \mid \text{Turing machine } M \text{ accepts some input in at most } t \text{ steps}\}$

(3) (30 points) The following special case of Post Correspondence Problem (PCP) is shown to be undecidable during lecture:

$$\text{PCP}' = \{\langle C\rangle \mid C \text{ is a finite collection of nonempty tiles with a match}\}\ ,$$

where a tile is nonempty if both its top and bottom strings are not the empty string.

Show that the following variants of PCP are undecidable, by reducing from PCP$'$ (over some alphabet Σ').

(a) PCP$_1$: PCP over the alphabet $\Sigma = \{\texttt{0}, \texttt{1}\}$.
The *alphabet* of PCP is the set of symbols that can appear on the tiles. For example, here is an instance of PCP over the alphabet $\Sigma = \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}\}$:

$$\boxed{\begin{array}{c}\texttt{ab}\\ \texttt{bc}\end{array}}\quad \boxed{\begin{array}{c}\texttt{ab}\\ \texttt{d}\end{array}}\quad \boxed{\begin{array}{c}\texttt{d}\\ \texttt{cd}\end{array}}\quad \boxed{\begin{array}{c}\texttt{b}\\ \texttt{d}\end{array}}$$

(b) PCP$_2$: PCP with alphabet permutation. Given finite alphabets Σ_T and Σ_B of the same size, a (Σ_T, Σ_B)-tile has a top string over Σ_T and a bottom string over Σ_B. A collection C of (Σ_T, Σ_B)-tiles has a match if there is a bijection $p : \Sigma_T \to \Sigma_B$, such that after mapping all the top symbols according to p, the new collection of tiles has a match (in the usual sense).

$$\text{PCP}_2 = \{\langle \Sigma_T, \Sigma_B, C\rangle \mid C \text{ is a finite collection of } (\Sigma_T, \Sigma_B)\text{-tiles with a match}\}\ .$$

For example, here is a collection of (Σ_T, Σ_B)-tiles, over the top alphabet $\Sigma_T = \{\texttt{A}, \texttt{B}\}$ and the bottom alphabet $\Sigma_B = \{\texttt{c}, \texttt{d}\}$:

$$\boxed{\begin{matrix}\texttt{AB}\\ \texttt{c}\end{matrix}} \quad \boxed{\begin{matrix}\texttt{BB}\\ \texttt{c}\end{matrix}} \quad \boxed{\begin{matrix}\texttt{B}\\ \texttt{dd}\end{matrix}}$$

This instance has a match (via the bijection $p(\texttt{A}) = \texttt{c}$ and $p(\texttt{B}) = \texttt{d}$):

$$\boxed{\begin{matrix}\texttt{AB}\\ \texttt{c}\end{matrix}} \quad \boxed{\begin{matrix}\texttt{B}\\ \texttt{dd}\end{matrix}}$$