Computability and Complexity

Department of Computer Science, University of Copenhagen,
Fritz Henglein & Christian Wulff-Nilsen

Written exam for 4 hours, April 14th, 2016

All usual aids such as books, notes, exercises, and the like may be used during the exam, but not your own calculators, computers, cell phones, or similar equipment. The exam must be answered in Danish or English.

This exam has 4 pages and consists of 3 questions, where Question 1 weighs approximately $\frac{1}{2}$ and Questions 2 and 3 each approximately $\frac{1}{4}$.

Question 1

Let $\Sigma = \{$`(`, `)`, `[`, `]`$\}$. A string over Σ is *balanced* if each opening parenthesis `(` and opening bracket `[` occurring in the string has a unique corresponding closing parenthesis `)`, respectively closing bracket `]`, to its right such that the substring between them is also balanced. Conversely, each closing symbol must have a corresponding unique opening symbol. For example, the strings `(()())`, `[([()])()]`, `[()]` and the empty string are balanced, but `(()()`, `())` and `[(])` are not balanced.

The *prefix nesting depth* of a prefix $p = a_1 \dots a_i$ of a given string $s = a_1 \dots a_n$ is the number of occurrences of opening symbols minus the number of closing symbols in p. The *nesting depth* of s is the maximum prefix nesting depth of any of its prefixes. For example, the nesting depth of `(()())` is 2 because the prefixes `((` and `(()(` have 2 more open parentheses than closing parentheses and no prefix has a higher difference of open and closed parentheses; the nesting depth of `[([()])()]` is 4 because prefix `[([(` has prefix nesting depth 4 and no prefix has a higher prefix nesting depth.

Define the following languages, each over alphabet Σ:

- $B = \{s \mid s \in \Sigma^* \wedge s \text{ is balanced}\}$, the balanced strings;
- $P = \{s \mid s \in B \wedge s \in \{$`(`, `)`$\}^*\}$, the balanced strings consisting of only parentheses;
- $P_3 = \{s \mid s \in P \wedge s \text{ has nesting depth } 3\}$, the balanced parenthesis strings with nesting depth exactly 3;
- $B_e = \{s \mid s \in B \wedge s$ contains the same number of `(` and `[` $\}$, the balanced strings with an equal number of parentheses and brackets.

Prove each of the following statements.[1]

Part 1.1 B is context-free, but not regular.

[1]If your answer includes a construction such as a grammar or a Turing Machine description, include an argument for its correctness.

Part 1.2 B_e is not context-free.

Part 1.3 P_3 is regular.

Part 1.4 P is in L.

Part 1.5 B is in L.

Hint: For $s \in \Sigma^*$, let $\overline{s}$ be s with each occurrence of `[` replaced by `(` and each `]` replaced by `)`. Note that $\overline{s} \in \{\texttt{(},\texttt{)}\}^*$. Use the following characterization of B: $s = a_1 \dots a_n \in B$ if and only if

1. $\overline{s} \in P$ and
2. for all $1 \leq i, j \leq n$, if $\overline{a_i} = \texttt{(}$ and $\overline{a_j} = \texttt{)}$ and $\overline{a_j}$ is the closing parenthesis corresponding to $\overline{a_i}$ then $a_i = \texttt{[}$ if and only if $a_j = \texttt{]}$.

Question 2

Let natural number d be given. Recall the definition of PCP in Section 5.2 of Sipser. We define a variant d-PCP as follows. An instance of d-PCP is a collection P of dominos

$$P = \left\{ \left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \ldots, \left[\frac{t_k}{b_k}\right] \right\},$$

and a d-match is a sequence $i_1, i_2, \ldots, i_l$ such that the two strings $t_{i_1} t_{i_2} \cdots t_{i_l}$ and $b_{i_1} b_{i_2} \cdots b_{i_l}$ have the same length and differ in at most d symbol positions.

Part 2.1 Show that for any natural number d, d-PCP is undecidable.

[Hint: repeat symbols a suitable number of times in an instance of PCP.]

Part 2.2 Define L_{erase} as the language consisting of descriptions of deterministic single-tape TMs M such that for each string w, when M is applied to w, M will either run forever or will in some execution step be in a configuration where all tape locations are blank and the tape head is in the left-most position (we do not require M to halt as soon as it reaches such a configuration). Equivalently, $\langle M \rangle$ belongs to L_{erase} if and only if M is a deterministic single-tape TM such that for every string w, if a computation history (accepting or rejecting) for M on w exists, this history contains a configuration where all tape locations are blank and the tape head is in the left-most position.

Show that L_{erase} is co-Turing-recognizable.

[Hint: consider all input strings in parallel.]

Part 2.3 Let $\langle M \rangle \in L_{erase}$ be given. Let S be the set of all possible tape contents that M can have when halting. Formally,

$$S = \{s \in \Sigma^* | \exists w \in \Sigma^* \text{ such that } M \text{ halts on input } w \text{ with } s \text{ on its tape}\}.$$

Show that $|S| \leq |Q|$ where Q is the set of states of M.

Question 3

Given natural number k, the k-HAMPATH problem is the following: given a directed graph G and two nodes s and t in G, is there a set of k or fewer paths in G such that

- each path in the set goes through each node of G at most once,
- each node of G belongs to precisely one path from the set, and
- one path from the set starts in s and ends in t?

Part 3.1 Let natural number k be given. Formulate k-HAMPATH as a language and show that it belongs to NP.

Part 3.2 Let natural number k be given. Show that k-HAMPATH is NP-complete.

Part 3.3 A ladder is a sequence of strings $s_1, s_2, \ldots, s_k$, wherein every string differs from the preceding one by exactly one character. For example, the following is a ladder of English words, starting with "head" and ending with "free":

head, hear, near, fear, bear, beer, deer, deed, feed, feet, fret, free

(in general, we do not require that a ladder consists of English words).

Let $LADDER_{DFA}$ be the language of strings $\langle M, s, t\rangle$ where M is a DFA and $L(M)$ contains a ladder of strings, starting with s and ending with t.

Show that $LADDER_{DFA}$ is in PSPACE.

[Hint: use Savitch's theorem.]

END OF EXAM SET