15-354: Midterm

Thursday, October 10, 2024

Instructions

- You have until Friday, 10/11, 24:00, to work on this test.
- The latex source is at midterm-24.tex. Your can type your answers directly into this file, but make sure to change the file name to `FirstnameLastname.tex`. To compile, you also need to download ks-exam.cls and ks-texmacs.sty.
- When you are done, email your pdf to me at `sutner@cs.cmu.edu`.
- You can consult all course materials, but no other sources.
- Do not talk to anyone about this exam.
- Post questions to ed, but make your posts private (unless it's about a typo or the like).
- Good luck.

Problem 1: Multiple Choice (24 pts.)

Answer each of the following questions **true** or **false**, no justification is necessary.

1. Wurzelbrunft thinks he has found a clone of total arithmetic functions that is somewhat difficult to describe (his paper is 120 pages long), but it captures intuitive computability perfectly. Could he be right?

   Answer:

2. Imagine someone hands you a BlackBox that somehow solves the Halting problem. Could one use this device to decide whether a computable function is total?

   Answer:

3. Suppose a set $A \subseteq \mathbb{N}$ is decidable. Is is it always true that there is a primitive recursive decision algorithm for A?

   Answer:

4. Suppose $L \subseteq \Sigma^*$ is an infinite regular language. Is it always possible to partition L into two infinite regular languages?

   Answer:

5. Is there an infinite language $L \subseteq \{a\}^\star$ that has no infinite regular sub-language?

   Answer:

6. Let $L \subseteq \{a\}^\star$ be an undecidable language. Is the Kleene star of L always undecidable?

   Answer:

Problem 2: Register Complexity (16 pts.)

Let us say a register R in a register machine is bounded if there is some fixed bound B such that during the computation on some arbitrary input the value stored in R is never larger than B. Two registers R_1 and R_2 are entangled if, at the end of every halting computation, they have the same value.

Justify your answer for both questions below.

A. Is it decidable whether a register is bounded?

B. Is it decidable whether two registers are entangled?

Problem 3: Wurzelbrunft Functions (30 pts.)

Someone has given Wurzelbrunft a recursion theory book on his birthday. He finds large function values produced by monsters such as the Ackermann function confusing, so he decides to study k-Wurzelbrunft functions: any total computable function $f : \mathbb{N} \to \mathbb{N}$ such that $f(x) < k$ for all x. Here $k \in \mathbb{N}$.

Wurzelbrunft thinks that, for sufficiently small k, all k-Wurzelbrunft functions are primitive recursive. Help him come up with an actual theorem and a proof. The stronger the theorem, the better.

A. Clearly state your k-Wurzelbrunft theorem.

B. Prove your theorem.

Problem 4: A-RMs (40 pts.)

Background

Fix some set $A \subseteq \mathbb{N}$. Suppose we modify a register machine by adding a an instruction `magicA k l` which executes as follows: the machine reads the number r in register R_0. Then it magically figures out whether r is in A; if so, execution continues in line k, and in line l otherwise.

Call these gizmos A-RMs.

Task

A. Show that all A-RMs can be simulated by an ordinary RM iff A is decidable.

B. For any A, construct a set H that cannot be decided by any A-RM.

C. Suppose you have access to a H-RM where H is the Halting set (for ordinary RMs). Explain how to exploit this to prove or disprove the Goldbach conjecture: every even number larger than 2 is the sum of two primes.

Problem 5: Computing Transients and Periods (20 pts.)

For the following, assume we are given a C program that computes a function $f : A \to A$ where $A = \{0, 1, \ldots, 2^{64} - 1\}$ (64-bit unsigned ints). Assume that the computation of $f(a)$ is expensive (though, by necessity).

Write t for the transient of a point in A, and p for the period. We can compute t and p in a memoryless way using Floyd's algorithm, but sometimes there are better solutions. If you write pseudo-code to describe your algorithms below, make sure to provide ample comments; no credit otherwise.

A. Suppose that the transients of all points under f are at most 3 (three). Give a fast and memory efficient algorithm to compute the transient and period of a point $a \in A$. State the running time of your algorithm in terms of t and p (try to spell out the constants).

B. Now suppose that the periods of all points under f are at most 3 (three). Give a fast and memory efficient algorithm to compute the transient and period of a point $a \in A$. State the running time of your algorithm in terms of t and p (try to spell out the constants).

Problem 6: Word Shuffle (30 pts.)

The word shuffle operation, in symbols $\parallel$, is a map from $\Sigma^* \times \Sigma^*$ to $\mathfrak{P}(\Sigma^*)$ defined by

$$\varepsilon \parallel y = y \parallel \varepsilon \;=\; \{y\}$$
$$xa \parallel yb = (x \parallel yb)\, a \cup (xa \parallel y)\, b.$$

As usual, we can extend the operation to languages by

$$K \parallel L = \bigcup \{\, x \parallel y \mid x \in K, y \in L \,\}$$

For example,

$$aa \parallel bbb = \{aabbb, ababb, abbab, abbba, baabb, babab, babba, bbaab, bbaba, bbbaa\}$$

Also, $(aa)^\star \parallel (bb)^\star$ is the set of all even/even words.

A. Given two words x and y, construct a finite state machine M that accepts $x \parallel y$.

B. Let $M_i = \langle Q_i, \Sigma, \delta_i; q_{0i}, F_i \rangle$, $i = 1, 2$, be two DFAs accepting regular languages L_i. Show how to construct a finite state machine M that accepts the shuffle language $L_1 \parallel L_2$. Make sure to specify the state set, the transitions, initial and final states.