15-354: Midterm

Thursday, October 12, 2023

Instructions

- You have until Friday, 10/13, 16:00, to work on this test.
- The latex source is at midterm-23.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. Suppose $A, B \subseteq \mathbb{N}$ and the symmetric difference between A and B is finite. Show that A is decidable whenever B is decidable.

   Answer:

2. Wurzelbrunft thinks he has a miraculous proof that every recursively enumerable set can be enumerated by a computable function f that is almost monotonic: $f(n) \geq f(m) - 42$ for all $n \geq m$. Is he right?

   Answer:

3. Suppose we compute a primitive recursive function on a register machine. Is the running time of the register machine always primitive recursive?

   Answer:

4. Let $L \subseteq a^\star$ be an infinite regular language. Is there always an infinite subset of L of the form $a^k(a^\ell)^\star$ for some $k \geq 0$, $\ell \geq 1$?

   Answer:

5. Does every infinite regular language $L \subseteq a^\star$ have an undecidable subset?

   Answer:

6. Suppose $L \subseteq a^\star$ is an infinite regular language. Are there always two disjoint infinite regular languages $K_1, K_2 \subseteq L$?

   Answer:

7. Suppose a set $A \subseteq \mathbb{N}$ is enumerated by a computable function. Is is it always true that the set can also be enumerated by a primitive recursive function?

   Answer:

8. Let $L \subseteq \Sigma^*$ be a decidable language. Is the Kleene star of L is also decidable?

   Answer:

Problem 2: Register Complexity (16 pts.)

Let us say a register R in a register machine is useful if it is an input/output register or there is some input $\boldsymbol{x}$ such that, during the computation on $\boldsymbol{x}$, the register R changes its value at least once. The whole machine is trim if all its registers are useful.

Justify your answer for both questions below.

A. Is it true that register machines with at most 1024 registers cannot implement all computable functions $\mathbb{N} \to \mathbb{N}$?

B. Is it decidable whether a register machine is trim?

Problem 3: Wurzelbrunft and Ochsenfiesl vs. Collatz (20 pts.)

Wurzelbrunft and Ochsenfiesl are two grad students at a little known university in a galaxy far, far away. They both are fascinated by the Collatz function $C(x) = x/2$ for x even, and $C(x) = 3x + 1$ otherwise. The Collatz Conjecture says that all orbits of $x \geq 1$ under C contain the point 1. This has been verified numerically for $x < 2^{68}$ and there are some interesting theorems about C, but the conjecture is wide open[1].

At any rate, Wurzelbrunft and Ochsenfiesl decided to study the complexity of the Collatz set

$$S = \{ x \in \mathbb{N}_+ \mid \text{ orbit of } x \text{ under } C \text{ contains } 1 \}.$$

Wurzelbrunft thinks he has a proof that S is decidable. He also claims that his result, together with the well-known fact that S is infinite, immediately implies the Collatz conjecture. Ochsenfiesl, on the other hand, thinks he has a proof that S is undecidable. He claims his result implies that the Collatz conjecture is false.

If you were their PhD advisor, what professional, well-reasoned advice would you give to them?

A. Wurzelbrunft:

B. Ochsenfiesl:

[1] Jeff Lagarias has written a book on the problem, and thinks it's harder than the Riemann hypothesis.

Problem 4: Speedy Iteration (20 pts.)

Suppose a function $f : [n] \to [n]$ is given as a lookup table, say, a plain array of integers. Think about n as being fairly large, somewhere between 2^{20} and 2^{30}. Clearly we can compute $f^t(x)$, $x \in [n]$, $t \geq 0$, by repeated lookup. This problem is about speeding up this computation.

The pre-computation in part (B) can store additional information, but make sure to use sub-quadratic space.

A. Explain how to reduce the problem of computing $f^t(x)$, to the problem of computing only values $f^{t'}(x)$ for $t' < n$.

B. Now assume that we need to perform many computations of various values $f^t(x)$. Show how to organize a pre-computation that speeds up these queries.

C. State clearly the cost of the pre-computation (time and space) and the improved evaluations.

Problem 5: Factors (20 pts.)

A word u is a factor of a word v if $v = xuy$ for some words $x, y \in \Sigma^*$. Write $\mathsf{fac}(L)$ for the language of all factors of a language L. For example, for the even/even language EE over $\{a, b\}$ we have $\mathsf{fac}(EE) = \{a, b\}^\star$.

In part (A), do not use any nondeterministic machines. Try to make your algorithms below as simple as possible, but don't worry about efficiency.

A. Let $\mathcal{A}$ be the minimal DFA for L. Explain how to construct a DFA for $\mathsf{fac}(L)$.

B. Let $\mathcal{B}$ be some NFA for L. Explain how to construct an NFA for $\mathsf{fac}(L)$.