15-354: Midterm

October 15, 2019

Name:

Andrew ID:

Instructions

- Fill in the box above with your name and your Andrew ID. **Do it, now!**
- Clearly mark your answers in the allocated space. If need be, use the back of a page for scratch space. If you have made a mess, cross out the invalid parts of your solution, and circle the ones that should be graded.
- Scan the test first to make sure that none of the 12 pages are missing. Some parts may be on the back of the page.
- The problems are of varying difficulty and are not necessarily sorted in order of increasing difficulty. You might wish to pick off the easy ones first.
- You have 80 minutes. Good luck.

1	20	
2	20	
3	20	
4	20	
5	20	
Total	100	

Problem 1: Busy Beaver (20 pts.)

Rado's Busy Beaver function β is a standard example of an arithmetic function that fails to be computable.

A. Give a brief definition of β (you can use register machines or Turing machines).

B. Assume you had a decision algorithm for Halting. Explain how to compute β using this algorithm.

Problem 2: Ackermann (20 pts.)

The Ackermann function is the prime example of a computable function that fails to be primitive recursive:

$$\begin{aligned} A(0, y) &= y^+ \\ A(x^+, 0) &= A(x, 1) \\ A(x^+, y^+) &= A(x, A(x^+, y)) \end{aligned}$$

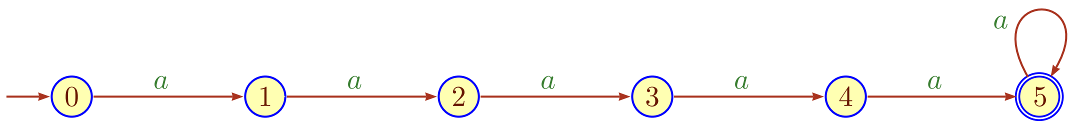
Here x^+ is an abbreviation for $x + 1$.

A. Show that for every fixed x, the function $A_x(y) = A(x, y)$ is primitive recursive.

B. How would you compute A on a register machine?

Problem 3: Hopcroft (20 pts.)

We have seen that the following type of DFA behaves very badly with respect to Moore's minimization algorithm. Here is $\mathcal{A}_5$:



A. What is the running time of Moore's algorithm on $\mathcal{A}_n$? Why?

B. By contrast, what is the running time of Hopcroft's algorithm on $\mathcal{A}_n$? Why?

Problem 4: Wurzelbrunft and Ochsenfiesl vs. Twins (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 prime twin conjecture (PTC): there are infinitely many primes p such that $p+2$ is also prime. Most number theorists agree that PTC is probably true, but currently no proof is available.

This is where Wurzelbrunft's and Ochsenfiesl's great and unmatched knowledge of computability theory comes into play. Consider the prime twin set

$$P = \{\, 0^p \mid p \text{ prime and } p+2 \text{ also prime} \,\} \subseteq \{0\}^{\star}.$$

Wurzelbrunft thinks he has found a DFA that accepts P. The machine is a bit large (some 10^{120} states), so he can't really write it down, but he is absolutely sure it works. Ochsenfiesl, in an enormous intellectual effort, has convinced himself that P is decidable.

A. Has Wurzelbrunft solved the PTC?

B. How about Ochsenfiesl?

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 $A = \{0, 1, \ldots, 10^{15} - 1\}$. Assume that the computation of $f(a)$ is constant time but expensive.

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).