15-354: Midterm

October 11, 2007

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 7 pages are missing. The problems are of varying difficulty and are not necessarily sorted according to 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: Counting and Decidability (20 pts.)

Suppose $A \subseteq \mathbb{N}$. Often it is important to understand the behavior of the corresponding counting function $f_A : \mathbb{N} \to \mathbb{N}$ defined by

$$f(n) = \text{ cardinality of } (A \cap \{1, 2, \ldots, n\})$$

For example, when A is the set of primes the counting function has been studied in great detail in number theory.

A. Show that f_A is computable when A is decidable.

B. Assume that A is semi-decidable. Show that A must be decidable when f_A is computable.

Problem 2: Iteration and Primitive Recursion (20 pts.)

Primitive recursive functions are obtained from simple basic functions by composition and primitive recursion. As we have seen, iteration is another important operation that produces useful functions. For $f : \mathbb{N} \to \mathbb{N}$ define its iterate

$$f^\dagger(n) = f^n(n)$$

A. Determine $f^\dagger$ when f is the successor function, $f(x) = x + 1$.

B. Show that $f^\dagger$ is primitive recursive whenever f is.

Problem 3: Selector Functions and Equivalence Relations (20 pts.)

The forward state merging algorithm is usually implemented using selector functions to represent equivalence relations on the state set $Q = \{1, 2, \ldots, n-1, n\}$: relation ρ is represented by an array R such that

$$R(x) = \min\big(z \mid z \; \rho \; x \big).$$

If you write pseudo-code to describe your algorithms below, make sure to provide ample comments; no credit otherwise. You can use any auxiliary data structure you like.

A. Given R, explain how to compute the number r of equivalence classes of ρ in time linear in n using $O(1)$ extra memory.

B. Given R and r, explain how to compute the size of a smallest equivalence class of ρ in time linear in n (there may be several smallest classes).

Problem 4: A Minimal Automaton (20 pts.)

For this problem, the input alphabet is $\Sigma = \{a, b\}$.

Consider the regular language $L = (bb)^\star aaa$.

A. Using the table method used in class, compute all the quotients of L. For your convenience, the table below has the first column filled in – but note that there may well be fewer than 8 quotients.

$-$	$(bb)^\star aaa$	L_1
$a^{-1}L_1$		
$b^{-1}L_1$		
$a^{-1}L_2$		
$b^{-1}L_2$		
$a^{-1}L_3$		
$b^{-1}L_3$		
$a^{-1}L_4$		
$b^{-1}L_4$		
$a^{-1}L_5$		
$b^{-1}L_5$		
$a^{-1}L_6$		
$b^{-1}L_6$		
$a^{-1}L_7$		
$b^{-1}L_7$		
$a^{-1}L_8$		
$b^{-1}L_8$		

B. Draw a *nice* picture of the resulting minimal automaton (make sure to indicate the initial state and all final states).

Problem 5: Alternative Union of Regular Languages (20 pts.)

Given two DFAs M_1 and M_2 for regular languages L_1 and L_2 we know how to construct a DFA $M_1 \times M_2$ for $L = L_1 \cup L_2$ using a product machine. Here is an alternative approach, exploiting non-determinism. Assume the two state sets are disjoint and define a new machine

$$M = \langle\, Q_1 \cup Q_2, \Sigma, \delta_1 \cup \delta_2, \{q_{01}, q_{02}\}\,, F_1 \cup F_2 \,\rangle$$

Note M has size $n_1 + n_2$ and is thus potentially quite a bit smaller than the product automaton. Of course, M is not a DFA.

A. Prove that M accepts L: you must argue about all possible runs of M on input x.

B. Describe the machine M' obtained by performing deterministic simulation on M. More precisely,

- Describe the state set Q' in terms of Q_1 and Q_2.
- Describe the transition function δ' in terms of δ_1 and δ_2.
- Describe the initial state.
- Describe the final states.