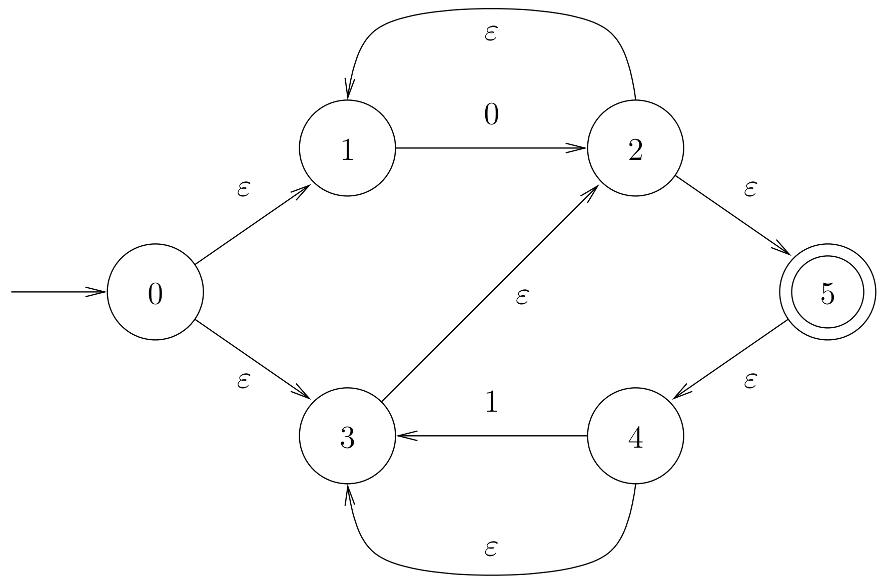
CSCE 551 Final Exam, Spring 2004
Answer Key

1. (10 points) Using any method you like (including intuition), give the unique minimal DFA equivalent to the following NFA:



If your answer is correct, you get full credit even if you do not show how you arrived at it.

Answer: The one-state DFA whose start start is also an accepting state, and both of whose transitions are self-loops. This DFA accepts every binary string, recognizing $\{0, 1\}^*$.

2. (10 points) Give an implementation-level description of a standard (1-tape deterministic) TM M that decides the following language over input alphabet $\{0, 1\}$:

$$\{w \mid w \text{ contains at least as many zeros as ones}\}.$$

Answer: "On input $w \in \{0, 1\}$:

(a) Scan right until a blank is encountered, replacing the first '1' seen with 'x'. If no '1' is seen in this scan, then accept.

(b) Scan left to the beginning, replacing the first '0' seen with 'x'. If no '0' is seen, then reject; otherwise, go to Step (a)."

Other algorithms are possible.

3. (10 points) Let A and B be two disjoint languages. Recall (Problem 4.18) that a language C *separates* A from B if $A \subseteq C$ and $B \subseteq \overline{C}$. Consider the two languages

$$A_{\mathrm{DIAG}} = \{\langle M \rangle \mid M \text{ is a TM that accepts the string } \langle M \rangle\}$$

and

$$R_{\mathrm{DIAG}} = \{\langle M \rangle \mid M \text{ is a TM that rejects the string } \langle M \rangle\}.$$

By filling in the bracketed parts, complete the following proof that there is no decidable language C that separates A_{DIAG} from R_{DIAG}:

> Suppose that there *is* some decidable C separating A_{DIAG} from R_{DIAG}. Let D be the following machine: "On input w: [YOU FILL IN THIS PART (HIGH-LEVEL DESCRIPTION)]." Consider D running on input $\langle D \rangle$. Clearly, D does not loop on input $\langle D \rangle$. But, if D accepts $\langle D \rangle$, then [YOU EXPLAIN WHY D MUST REJECT $\langle D \rangle$]. Likewise, if D rejects $\langle D \rangle$, then [YOU EXPLAIN WHY D MUST ACCEPT $\langle D \rangle$]. This is a contradiction, thus there is no such decidable C.

Answer: Here is a description of D. In fact, D decides $\overline{C}$.

$D :=$ "On input w:

(a) If $w \in C$, then reject; otherwise accept."

If D accepts $\langle D \rangle$, then $\langle D \rangle \notin C$ (by the definition of D), and thus D does not accept $\langle D \rangle$ (by the definition of C), and thus D rejects $\langle D \rangle$ (because D is a decider). Likewise, if D rejects $\langle D \rangle$, then $\langle D \rangle \in C$ (by the definition of D), and thus D does not reject $\langle D \rangle$ (by the definition of C), and thus D accepts $\langle D \rangle$ (because D is a decider).

4. (10 points) Let f be any computable function. Show that the set

$$R = \{y \mid (\exists x \in A_{\mathrm{TM}})\ f(x) = y\}$$

is Turing-recognizable by giving a high-level description of a TM that recognizes R.

Answer: The following machine recognizes R:

$M :=$ "On input y:

(a) Cycling through every string x:
    i. Compute $f(x)$.
    ii. If $f(x) = y$, then accept; else continue to the next x."

5. (10 points) Find a mapping reduction from A_{TM} to the language

$$\{\langle M \rangle \mid M \text{ is a TM and } L(M) = A_{\mathrm{TM}}\}.$$

Answer: Let L be the language above. Fix a TM M_0 that loops on all inputs. (Thus $L(M_0) = \emptyset \neq A_{\mathrm{TM}}$, and so $\langle M_0 \rangle \notin L$.)

Let $f :=$ "On input x:

(a) If x is not of the form $\langle M, w \rangle$, where M is a TM and w an input string to M, then output $\langle M_0 \rangle$. // This works because $x \notin A_{\mathrm{TM}}$.

(b) Otherwise, we have $x = \langle M, w \rangle$ as above. Let R be the following TM:
$R :=$ 'On input $\langle N, y \rangle$, where N is a TM and y a string:

  i. Run M on input w.

  ii. If M ever accepts w, then run N on input y (and do what N does).

  iii. Otherwise, loop.'

(c) Output $\langle R \rangle$."

If M does not accept w, then $L(R) = \emptyset \neq A_{\mathrm{TM}}$. Conversely, if M does accept w, then $L(R) = A_{\mathrm{TM}}$. Thus

$$\langle M, w \rangle \in A_{\mathrm{TM}} \iff f(\langle M, w \rangle) \in L .$$

6. (10 points) Show that there is no computable function f outputting natural numbers such that, for any TM M and string w, if M accepts w, then M accepts w in at most $f(\langle M, w \rangle)$ steps. [Hint: Argue by contradiction.]

   Answer: Suppose there exists such an f. Then the following TM clearly decides A_{TM}:
   $D :=$ "On input $\langle M, w \rangle$, where M is a TM and w is a string:

   (a) Compute $t := f(\langle M, w \rangle)$.

   (b) Run M on input w for t steps.

   (c) If M accepts w within this time, then accept; else reject."

   This is a contradiction, because A_{TM} is undecidable.

7. Below, G is always an undirected graph, and k is a natural number. A path in G is *simple* if no vertex appears more than once along the path. The *length* of a path is the number of edges in the path.

   (a) (10 points) Explain why the language

   $$\mathrm{LONGPATH} = \{\langle G, v, k \rangle \mid G \text{ has a simple path of length } k \text{ starting at vertex } v\}$$

   is in NP.

   (b) (10 points) Show that if LONGPATH $\in$ P, then there is a polynomial-time computable function f that, on input $\langle G, k \rangle$, either outputs some simple path in G of length k or outputs "no" if there is no such path. You may take the statement of part (a) as given. A high-level description of f is fine.

Answer:

(a) LONGPATH is in NP because if G does have a simple path of length $\geq k$ starting at v, then a ptime verifiable proof could be such a path p itself, given as a sequence of vertices. The verifier checks that: there are at least k vertices in p; no vertex is repeated in p; p starts with v; and each vertex in p except the last is adjacent to the immediately following vertex in p.

(b) Here is a description of f:
"On input $\langle G, k \rangle$, where G is a graph and k a natural number:

i. Run through the vertices v of G, checking whether $\langle G, v, k \rangle \in$ LONGPATH.
ii. If there is no vertex v such that $\langle G, v, k \rangle \in$ LONGPATH, then output "no" and halt.
iii. Otherwise, let v be the first vertex found such that $\langle G, v, k \rangle \in$ LONGPATH.
iv. Initialize p to be the length 0 path consisting of just v.
v. While $k > 0$, do the following:
    A. Let G' be the graph obtained by removing v and its incident edges from G.
    B. Run through the neighbors[1] of v in G until a neighbor w is found such that $\langle G', w, k-1 \rangle \in$ LONGPATH. // Such a w must exist.
    C. Append w onto the end of p.
    D. Set $G := G'$ and $k := k - 1$.
vi. Return p."

The function f can be computed in ptime, because there are at most $2|V(G)| - 1$ calls to LONGPATH, each on a polynomial sized input. The rest of the algorithm (besides the calls to LONGPATH clearly takes polynomial time.

8. (10 points) Using any method you like, show that the language

$$\{0^m 1^n \mid m, n \geq 0 \text{ and } n \neq m^2\}$$

is not regular.

Answer: Let L be the language above. There are (at least) two different solutions to this problem:

Solution 1: Suppose L is regular. Then by the closure properties of regular languages, the language $L' = \overline{L} \cap L(0^*1^*)$ is also regular. But

$$L' = \{0^m 1^{m^2} \mid m \geq 0\} ,$$

[1]A *neighbor* of a vertex v is any vertex adjacent to v.

and this language cannot be regular, as is seen via the pumping lemma for regular languages: given any $p > 0$, set $s := 0^p 1^{p^2}$. Clearly $s \in L'$ and $|s| \geq p$. Given any x, y, z such that $xyz = s$, $|y| > 0$, and $|xy| \leq p$, it must be that $y = 0^k$ for some $k > 0$. Let $i := 0$. Then $xy^iz = xz = 0^{p-k}1^{p^2} \notin L'$. Thus L' is not regular. Contradiction. It follows that L cannot be regular.

Solution 2: Using the pumping lemma directly, given any $p > 0$, set $s := 0^p 1^{(p!+p)^2}$. Clearly, $|s| \geq p$ and $s \in L$, since $(p!+p)^2 \neq p^2$. Given strings x, y, z such that $xyz = s$, $|y| > 0$, and $|xy| \leq p$, it must be that $y = 0^k$ for some $0 < k \leq p$. Set $i := \frac{p!}{k} + 1$ (which is an integer). Then

$$xy^iz = 0^{p!+p}1^{(p!+p)^2} \notin L .$$

Thus L is not regular.