Open book, open notes, closed neighbors, 48 minutes. The exam totals 80 pts., subdivided as shown. Recall that \(#0(x)\) stands for the number of 0s in \(x\), \(#1(x)\) for the number of 1s in \(x\), and \(D_{TM}\) for the diagonal language \(\{e : e \notin L(M_e)\}\).

(1) \((5 \times 5 = 25\) pts.) **Multiple Choice.**

Write the best answer (a,b,c,d) in the exam book provided. No justifications are needed, but they could possibly help for partial credit in case of a misunderstanding.

1. Given a regular expression \(r\), to obtain a regular expression \(r'\) such that \(L(r')\) equals the complement of \(L(r)\) (in \(\{0,1\}^*\) that is), one can:
   
   (a) Convert \(r\) to an NFA \(N\), create the NFA \(N'\) by interchanging the accepting and rejecting states of \(N\), and convert \(N'\) to \(r'\).
   
   (b) Convert \(r\) to an NFA \(N\), convert \(N\) to a DFA \(M\), create the DFA \(M'\) by interchanging the accepting and rejecting states of \(M\), and convert \(M'\) to \(r'\).
   
   (c) Change every 0 appearing in \(r\) to 1 and change every 1 to 0.
   
   (d) Repeatedly apply DeMorgan’s Law: \(\sim (r_1 + r_2) = (\sim r_1) \cap (\sim r_2)\).

2. If \(A \leq_m B\) and \(A\) is neither c.e. nor co-c.e., then:
   
   (a) \(B\) is decidable.
   
   (b) \(B\) is c.e. but might not be co-c.e.
   
   (c) \(B\) is neither c.e. nor co-c.e.
   
   (d) any one of (a,b,c) might be true—we cannot tell.

3. \(\emptyset \cdot \emptyset^*\) equals which of the following—?
   
   (a) \(\emptyset\).
   
   (b) \(\lambda\)
   
   (c) \(\{\lambda\}\)
   
   (d) \(\{e : L(M_e) = \emptyset\}\), i.e., \(E_{TM}\).

4. If \(A\) is the language of a deterministic PDA that runs in real time, then:
   
   (a) \(A\) is in the class \(P\).
   
   (b) \(A\) is accepted by a one-tape TM that runs in quadratic time. (cont’d overleaf)
   
   (c) \(A\) could be regular or non-regular—we don’t know which.
   
   (d) all of the above.

5. If \(A\) is a regular language but no string beginning with 000 belongs to \(A\), then:
(a) no string beginning with 111 belongs to $A$ either.
(b) a Myhill-Nerode proof involving $A$ must start by taking $S = 1^*$ not $S = 0^*$.
(c) there is a DFA $M$ such that $L(M) = A$ and $M$ processes 000 from its start state to a dead state.
(d) the complement of $A$ equals $000(0 + 1)^*$.

Answer: 1.(b), 2.(c), 3.(a), 4.(d), 5.(c). The answer 1.(a) is wrong in general because complementing $F$ in an NFA does not necessarily complement the language. Option 1.(c) bit-complements every string in the language. Option 1.(d) is not enough to finish the job. Option 3.(c) would be right if the given were just $\emptyset^*$, but the leading “$\emptyset$” kills it. In 5.(d) you only know that the complement contains $000(0 + 1)^*$.

(1′) (1 pt.)

Name a class of languages that is not closed under complementation.

Answer: RE is an example, co-RE is another. It is commonly believed but not known that NP is not closed under complements. Another class that definitely isn’t—but the reason is not covered in this course—is the class of languages accepted by nondeterministic PDAs, which equals the class CFL of context-free languages.

(2) (18 + 9 = 27 pts.)

Let $N = (\{1, 2, 3\}, \{a, b\}, \delta, 1, \{1\})$ be the NFA with
\[ \delta = \{(1, \lambda, 2), (1, b, 3), (2, a, 3), (2, b, 2), (2, b, 3), (3, b, 1)\}. \]
(Yes, there are two arcs on $b$ out of state 2 and arcs on both $a$ and $b$ going from 2 to 3.)

(a) Calculate a DFA $M$ such that $L(M) = L(N)$.
(b) Calculate a regular expression $r$ such that $L(r) = L(N)$.

Answer: We have “whenever 1 then 2” by the $\lambda$-arc, so the start state $S$ of the DFA $M$ equals $\{1, 2\}$ not $\{1\}$. I like the stepping-stone of computing the extended delta function:
\[
\tilde{\delta}(1, a) = \emptyset \quad \tilde{\delta}(1, b) = \{3\} \\
\tilde{\delta}(2, a) = \{3\} \quad \tilde{\delta}(2, b) = \{2, 3\} \\
\tilde{\delta}(3, a) = \emptyset \quad \tilde{\delta}(3, b) = \{1, 2\}
\]
Then the rule $\Delta(P, c) = \cup_{p \in P} \tilde{\delta}(p, c)$ takes over and we get:
\[
\Delta(S, a) = \tilde{\delta}(1, a) \cup \tilde{\delta}(2, a) = \emptyset \cup \{3\} = \{3\} \\
\Delta(S, b) = \tilde{\delta}(1, b) \cup \tilde{\delta}(2, b) = \{3\} \cup \{2, 3\} = \{2, 3\}. \\
\Delta(\{3\}, a) = \tilde{\delta}(3, a) = \emptyset \\
\Delta(\{3\}, b) = \tilde{\delta}(3, b) = \{1, 2\} \\
\Delta(\{2, 3\}, a) = \tilde{\delta}(2, a) \cup \tilde{\delta}(3, a) = \{3\} \\
\Delta(\{2, 3\}, b) = \{2, 3\} \cup \{1, 2\} = \{1, 2, 3\} \text{ (which is new)} \\
\Delta(\{1, 2, 3\}, a) = \{3\} \\
\Delta(\{1, 2, 3\}, b) = \{1, 2, 3\}.
\]
and finally done since we know the dead state goes to itself. All five states—two accepting and three rejecting—are needed.

(b) The easiest is to bypass state 2, which changes the arc from 1 to 3 from $b$ to $b + b^*(a + b)$. This incidentally simplifies to just $b^*(a + b)$. Now we have a 2-state GNFA $T$ with no self-loops at state 1 or 3, so the language is new-$T(1, 1) = [T(1, 3)T'(3, 1)]^* = (b^*(a + b)b)^*$. Also correct are $(b^*(ab + bb))^*$ and the redundant $(bb + b^*(ab + bb))^*$; the latter is what you might get if you were to bypass state 3 instead.

**Do Exactly One of the following two problems:**

(3a) (27 pts. total)

Define $L = \{ x0y : \#0(x) = \#1(y) \}$. Is $L$ regular? Prove your answer.

**Answer:** $L$ is non-regular—the ‘0’ in $x0y$ makes a difference compared to $\{ xy : \#0(x) = \#1(y) \}$, which equals $(0 + 1)^*$. For an MNT proof, take $S = 0^+$. Clearly $S$ is infinite. Let any $x, y \in S$, $x \neq y$, be given. Then we can write $x = 0^m$ and $y = 0^n$ where wlog. $m < n$. Take $z = 1^{n-1}$. Then $yz = 0^m1^{n-1} = 0^{n-1} \cdot 0 \cdot 1^{n-1} \in L$, but $xz = 0^m1^{n-1} = 0^{n-1}01^{n-1}$. Note that by taking $S = 0^+$ rather than $S = 0^*$ we can talk about “$0^{n-1}$” without having to make an exception in case $x = \lambda$. Because the other breakdowns of $xz$ into “$x0y$” are even more unfavorable and $n$ is too big, we have $xz \notin L$ as needed. So $L(xz) \neq L(yz)$, and since $x, y$ are arbitrary distinct members of $S$, $S$ is PD for $L$; since $S$ is infinite, $L$ is non-regular by the Myhill-Nerode Theorem.

(3b) (27 pts. total)

Define $A = \{ e : M_e$ does not accept any palindrome $\}$. Show by reduction from $D_{TM}$ that $A$ is not c.e. (If you wish, you may work with the complements of $A$ and $D_{TM}$ in your reduction instead, and may regard $e$ as the ASCII text of a program rather than the code/Gödel-number of a Turing machine.)

**Answer:** $\overline{A} = \{ e : M_e$ does accept some palindrome $\}$. The “all-or-nothing-switch” in class automatically works to give $K \leq_m \overline{A}$: Map any given TM $M$ to a TM $M_e$ that on any input $w$ simulates $M(M)$ and accepts $w$ iff it found that $M$ accepted its own code. Then $M \in K \implies L(M_e) = \Sigma^*$ which certainly includes some palindromes, but $M \notin K \implies L(M_e) = \emptyset$ which doesn’t include any. Clearly the code map $f(M) = M_e$ is computable. Complementing the languages on both sides of the reduction gives $D \leq_m A$ via the same $f$.

END OF EXAM.