Open book, open notes, closed neighbors, 48 minutes. The exam totals 80 pts., subdivided as shown. Show all work—this may help for partial credit. Please do the problems on these exam sheets.

All notation is standard as in course readings and lectures; in particular, the reversal of string $x$ is written $x^R$, with examples $011^R = 110$, $\epsilon^R = \epsilon$, and $x = x^R$ means $x$ is a palindrome. the difference of two sets $B$ and $A$ is written $B \setminus A$; it also equals $B \cap \bar{A}$ where $\bar{A}$ stands for the complement of $A$.

(1) (24 pts.)

Over $\Sigma = \{0, 1\}$, define $L$ to be the language of strings that have more occurrences of the substring 010 than occurrences of the substring 101. Prove using the Myhill-Nerode technique that $L$ is not a regular language. (Note that occurrences of the substrings can overlap—but you can take that worry out of the picture by judicious choices of $S$ and/or $z$.)

Answer: Take $S = (010)^*$, which is clearly infinite. Let any $x, y \in S (x \neq y)$ be given. Then there are numbers $m, n \geq 0$ such that $x = (010)^m$ and $y = (010)^n$, where $m < n$ wlog. Take $z = 01(101)^m$. Then $xz = (010)^m01(101)^m$ has $m$ occurrences of the substring 010, because repetitions like 010010 do not have any occurrences in the “overlaps” and 101101 does not have any occurrence of 010 at all. Likewise, $xz$ has exactly the obvious $m$ occurrences of the substring 101, but no fewer—so it is not in $L$. Whereas, $yz = (010)^m01(101)^m$ has $n$ occurrences of 010, more than the number $m$ of occurrences of 101, so $yz$ is in $L$. Thus $L(xz) \neq L(yz)$, and since $x, y \in S$ are arbitrary, $S$ is PD for $L$ as well as infinite, so $L$ is nonregular by MNT.

Variations: In fact, is it OK to omit the extra 01 in the middle. Except for $m = 0$ (which you can rule out ahead of time by taking $S = (010)^+$ instead), the juncture between $x$ or $y$ and $z$ has 010101. This gives one extra 010 but also one extra 101, so they offset and the inequalities stay the same. Provided you noticed the extra occurrences, that was fine. By symmetry, the choice $S = (101)^*$ works in much the same way, with optional middle part 10 instead, and you could even “proactively prophylactically” incorporate it into $S$ as $(101)^*10$.

Half noticed the overlaps issue, half did not. The problem was designed to be scored “forgivingly” for the latter. Some made the “too many stars” mistake, with things like $S = 0^*(010)^*$. Some dropped parens in writing e.g. $010^m$ and $101^m$ but argued as if it were the correct $(010)^m$ and $(101)^m$ anyway.

(2) (9 + 15 + 6 + 2 = 32 pts.)

Consider the regular expression $r = (aa + ba)^*(ab + bb)^*$ over the alphabet $\Sigma = \{a, b\}$.

(a) Design an NFA $N$ with 4 states such that $L(N) = L(r)$.

(b) Design a DFA $M$ such that $L(M) = L(r)$. You may either convert your NFA $N$ from part (a) or design $M$ by understanding $r$ directly, but in the latter case you must have comments that explain the strategy. In particular, you must have at least one comment saying when the $(ab + bb)^*$ part comes into play.

(c) Does there exist a “kill string” $u$ such that for all $z \in \Sigma^*$, $uz \notin L(r)$? Give a shortest such string if so.
(d) Convert your \( N \) into a 2-state generalized NFA \( G \) such that \( L(G) = L(N) \). You need not follow the algorithm but can “just do it” — and this is worth 2 points mainly because 80 is congruent to 2 modulo 3.

Answer: (a) An NFA \( N \) that follows the structure of \( r \) is at upper left in the diagram below. A similar NFA without the \( \epsilon \)-transition is at upper right. (It was also possible to design the DFA as in part (b) and then remove the dead state to call it an NFA.)

(b) The DFA obtained via the set-of-states construction from the first NFA is at lower left. This comes about from

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>( a )</th>
<th>( b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s )</td>
<td>{2}</td>
<td>{2}</td>
</tr>
<tr>
<td>2</td>
<td>{s, f}</td>
<td>\emptyset</td>
</tr>
<tr>
<td>( f )</td>
<td>{3}</td>
<td>{3}</td>
</tr>
<tr>
<td>3</td>
<td>\emptyset</td>
<td>{f}</td>
</tr>
</tbody>
</table>

The start state is \( \{s, f\} \) not just \( \{s\} \). The NFA at upper right gives a similar construction except that the start state is just \( \{s\} \) (which is an accepting state) and has \( \delta(s, a) = \{2, 3\} \), \( \delta(s, b) = \{2, 3\} \) too. Both yield the same DFA except for the label of the start state, which is also the minimum DFA for the language.

(c) Any string in \( M \) that takes a shortest route to the dead state works: \( abaa \), \( bbba \), \( abba \), \( bbba \) all do it.

(d) The first 2-state GNFA shown basically just takes the two-state loops of the NFA \( N \) and makes them single loops. The second one uses the entire regular expression \( r \) in a single edge from \( s \) to \( f \); this is “cheesy” but complies with the condition.
The most common error was confusing \( r \) with \((aa + ba + ab + bb)^*\), which has no “sequentiality” and hence no dead condition as with \(abba\) etc. This was not expected to be so hard a problem. On part (d), a couple people did give the latter answer or one like it.

(3) (24 pts. total) True/False with justifications: each question is 3 pts. for correct answer and 3 for an explanation (or definitive example or counterexample) in one or two sentences.

(a) If \( x \) is any string in the language \( L(r) \) in problem (2), then its reversal \( x^R \) is also in \( L(r) \).

(b) Every undecidable language is non-regular.

(c) If \( B \) is undecidable and \( A \) is decidable, then \( B \setminus A \) must be undecidable.

(d) If \( B \) is non-regular and \( A \) is regular, and if \( A \subset B \), then \( B \setminus A \) must be non-regular.

Answers: (a) False: \( abaa \) is not in \( L(r) \) (in fact, it is a kill string) but its reversal is \( aaab \) which belongs to \( L(r) \).

(b) True: Every regular language is decided by a DFA, and this is the contrapositive.

(c) False: A possible case has \( A = \Sigma^* \), which is decidable (in fact, regular). But then for any language \( B \) whatever, \( B \setminus A = \emptyset \), which is also decidable (and regular).

(d) True: Having the extra condition \( A \subset B \) rules out taking \( A = \Sigma^* \), and more in particular, assures that \( B \) is the union of \( A \) and \( B \setminus A \). If \( B \setminus A \) were regular, then since \( A \) is regular, this would make \( B \) the union of two regular sets, and hence regular, a contradiction. So \( B \setminus A \) must be non-regular.

Several people did get 3(a) perfectly even while having messed up on problem (2). The grading is giving benefit of doubt on cases where the answer to 3(a) is wrong but would be correct for the NFA and DFA given, such as if they had \((aa + ba + ab + bb)^*\) as the language. Only a few got the reasons for (c) and (d) completely correct.