Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 1. (15 points) Prove, using the pumping lemma for context-free languages, that the language of all palindromes over the alphabet $\{0,1\}$ in which the numbers of 0s and 1s are equal, is not context free.
Note: We will grade this problem very closely, so make sure that your argument is complete and that no details are left implicit. The problem is not hard, the reason for this exercise is for you write a complete and precise proof.

Let p be the pumping length the pumping lemma guarantees. The string s = $0^p1^{2p}0^p$ is within the language of palindromes. Let s = uvxyz, where |vy| > 0 and |vxy| <= p. These conditions give us the following five cases.

1) vxy is contained with the first set of 0s. When the string is pumped up, the number of 0s within the first half of the string will be greater than in the second half and so the string will not be a palindrome and thus is not within the language.
2) vxy is contained within the 1s in the middle of the string. When the string is pumped up, the number of 1s will increase while the number of 0s will stay the same. As such, the number of 0s and 1s will not be equal and so the string is not within the language.
3) vxy is contained within the second set of 0s. When the string is pumped up, the number of 0s within the second half of the string will be greater than in the first half and so the string will not be a palindrome. Thus it is not within the language.
4) vxy contains 0s from the the first part of the string and 1s from the middle of the string. When the string is pumped up, the number of 0s in the first half and the number of 1s increase. As a result, there is a greater number of 0s in the first part than the last and additionally, the number of 0s don’t equal the number of 1s. As such, the string is not within the language.
5) vxy contains 1s from the middle of the string and 0s from the last part of the string. When the string is pumped up, the number of 1s and the number of 0s in the last part of the string increase. As a result, there is a greater number of 0s in the last part than the first and additionally, the number of 0s don't equal the number of 1s. As such, the string is not within the language.

Since in every case, pumping the string s results in a string not in the language, this contradicts the pumping lemma for CFLs and therefore the language of palindromes is not context-free.

Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 2. (10 points) Show that the class of TM-decidable languages is closed under the following operations: union, concatenation, star, intersection, and complement.

Union
Define a Turing Machine M that will simulate machine M_1 and M_2 where M_1 will accept language A and M_2 language B. If a string w simulated in M_1 or M_2 is accepted in one or both of these machines, then the turing machine M accepts the string. Otherwise, the turing machine M will reject the string w.

Concatenation
Define a Turing Machine M that will simulate machine M_1 and M_2 where M_1 will accept language A and M_2 language B. If the string w can be broken into two parts where the first half is accepted by M_1 and the second half accepted by M_2, then it will be accepted by M, otherwise M rejects it.

Star
Turing machine M accepts A*. We will take the string w and break into parts where $w=x_1x_2...x_n$. If every part of w can be accepted by M_A which is simulated by M, then the input is accepted by the turing machine M. Otherwise, it will reject.

Intersection
Define a Turing Machine M that will simulate machine M_1 and M_2 where M_1 will accept language A and M_2 language B. If a string w simulated in M_1 and M_2 is accepted in both of these machines, then the turing machine M accepts the string. Otherwise, the turing machine M will reject the string w.

Complement
Define a Turing Machine M that will simulate machine M_1 for a language A. If a string w simulated in M_1 is rejected, it will be accepted by M. Inversely, if a string w that is simulated by M_1 is accepted, it will be rejected by M.

Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 3. (10 points) Show that the class of TM-recognizable languages is closed under the following operations: union, concatenation, star, and intersection. Is it closed under complement?

<u>Union</u>
Define a Turing Machine M that will simulate machine M_1 and M_2 where M_1 will accept language A and M_2 language B. Alternate between simulating a step on M_1 and then M_2. If on a given step, M_1 and/or M_2 accepts then the Turing machine M accepts the string.

<u>Concatenation</u>
Define a Turing Machine M that will simulate machine M_1 and M_2 where M_1 will accept language A and M_2 language B. Alternative between simulating steps on M_1 and M_2. If either M_1 or M_2 eventually accept the string, then the Turing machine M accepts the string.

<u>Star</u>
Define a Turing Machine M and language A, where M accepts A*. Let w be a string of language A*. We take w and break into parts such that $w = x_1x_2...x_n$. If for every x it is accepted in a simulated machine from M, M_A, up until n, then M accepts the string. Otherwise, it can go on forever, or reject if w is not within the language.

<u>Intersection</u>
Define a Turing Machine M that will simulate machine M_1 and M_2 where M_1 will accept language A and M_2 language B. Alternative between simulating steps on M_1 and M_2. If a string w is eventually accepted by both M_1 and M_2 then the Turing Machine M accepts it.

<u>Complement</u>
The class of TM-recognizable languages is not closed under complement. If a language and its complement are both TM-recognizable then the language is also TM-decidable. However, while every language that is TM-decidable is also TM-recognizable, the inverse is not true. As such, it is not possible for every language to be TM-decidable so the class of recognizable languages is not closed under complement.

Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 4. (10 points) Show that every infinite TM-recognizable language has an infinite decidable subset.

Enumerator: E
Language: A
Turing Machine: M
String: w

Let M be the Turing Machine that recognizes an infinite language A. Let E be the enumerator that prints out A that follows the following program.
E: For step i :

  Simulate M once on w_i where w_i is the ith string that A outputs
  Simulate M on every string w_j where $j<c$.
  Accept w_j if M accepts
  Reject w_j if M rejects

Since A is an infinite language, the E prints out an infinite subset of the given language.

we were really stumped and really burnt out after a long week, sorry