Jessica Noel & Jack Schniederhan

Problem Set 8

Problem 1

Prove that the language of all palindromes over the alphabet $\{0, 1\}$ in which the numbers of 0's and 1's are equal in not CF

Assume this language L is context free. Let p be the pumping length of L. Let $s = 0^p 1^{2p} 0^p$

By the pumping lemma, these conditions must be satisfied:

1) $|vxy| \leq p$ 2) $|vy| > 0$ 3) $uv^ixy^iz \notin L$

Three major conditions:

1) vxy is made of just 1s in the middle (not all, as $|vxy| \leq p$). In this case the number of 1's to go up without changing number of 0's, causing the string to leave the language.

2) vxy is made of just 0's from the first "section" or just 0's from the second "section". In this case, this will cause the string to be lopsided when pumped up, as 0's on one half of the string will be a different number than the other half. This means the string is not a palindrome, so we are no longer in the language

3) vxy is made of some 0's from first section and some 1's, OR some 0's from the second section and some 1s.

Visualization of this:

$$\underbrace{00\ldots0}_{p}\;\underbrace{2222\ldots2}_{2p}\;\underbrace{00\ldots0}_{p} \qquad (vxy \text{ spanning the end of the first } 0\text{s and the start of the } 1\text{s})$$

$$\underbrace{00\ldots0}_{p}\;\underbrace{1111\ldots1}_{2p}\;\underbrace{00\ldots0}_{p} \qquad (vxy \text{ spanning the end of the } 1\text{s and the start of the last } 0\text{s})$$

- In these cases, pumping vxy up will cause the string to no longer be a palindrome, as 0s on one side will differ from those on the other, causing the same lopsided effect as case 2.
- All cases are covered, so pumping s results in a string not in the language everytime. This contradicts the pumping lemma for CFLs; therefore L is not context free.

Problem 2

Show that the Class of TM-decidable language is closed under union, concatenation, star, intersection and complement.

- For all of the following parts let L_1 and L_2 be decidable languages and M1 and M2 the machines which recognize L1 and L2.

Union: Using a two-tape turning machine, M_1 follow the process $L = (L_1 \cup L_2)$

1) on input x, copy input x to the second tape
2) on the first tape run M_1 on x
3) If M_1 accepts, then accept for M
4) Else, on the second tape, run M_2 on x
5) If M_2 accepts, then accept for M else reject

Since, this makes M a decider and $L = L_1 \cup L_2$, they are closed under union.

Concatenation: using a two tape turning machine M_1, follow the process ($L = L_1$ concat L_2)

1) on input x, split the string into two halves $x = x_1 x_2$
2) Copy x_1 on first tape & x_2 on second tape
3) on Second tape run M_1 on x_1, if M_1 accepts then accept for M
4) else, on first tape run M2 on x_2. If accepted by M_2, accept M.
else reject.

Since M is a decider and L_1 is a concatenation of L_1 and L_2 they are closed under concatenation

Star: Using a two tape turning machine follow the process: ($L = L_1$*)

1) on input x, copy the most recent left part of the string that hasn't been read yet and copy onto the second tape.
2) Run M_1 on the second tape
3) If M_1 accepts and the string has been entirely processed accept for M.
4) else, if M_1 accept and the string has "stuff" left, loop back to step 1.

- M_1 is a decider and L^* is input, so its closed under star.

Prob 2 continued

Intersection: using a 2-tape turning machine, follow the process: ~~[illegible]~~ (L is $L_1 \cap L_2$)

1) On input x, copy x to the second tape

2) On first tape, run M_1 on x if M_1 rejects, reject

3) else, on second tape, run M_2 on x, if M_2 accepts, accept

4) else reject

- Since M is a decider and $L = L_1 \cap L_2$, they are closed under intersection

Complementation: Using a turning machine, follow the process: ($L = \overline{L_1}$)

1) On input x, run M_1 on x

2) If M_1 accepts, reject else accept

- This is backwards from the rest since the complement of L is the complete opposite of L. Since M is a decider and $L = L_1$ they are closed under complementation.

Problem 3

Show that the class of TM-recognizable languages is closed under union, concatenation, star, intersection, and complement.

Union: L1 and L2 are recognizable languages. M1 and M2 are machines which recognize L1 and L2. If we present the string X to M1 and it accepts X, we simply just accept it. If M1 is to reject string X we can run it through M2 which will either accept it or reject it. We can run M1 and M2 one at a time (alternate), this allows us to know that either M1 OR M2 can accept X, allowing a TM-recognizable language to be closed under union.

Concatenation: Again, let L1 and L2 be TM recognizable languages and M1 and M2 be machines that recognize this language. If we have an input string X and break it into X_1, X2. When we run these through M_1 (run X_1) and M_2 (run X_2) if both machines accept, then it is accepted and we know they are closed under concatenation.

Star: L will be our TM recognizable language and M the machine which recognizes it. If L is broken into m parts and we run all m's through machine M_1 if M accepts all m then we know L is accepted by M.

Intersection: If we have L1 and L2 as TM recognizable languages and M1 and M2 as machines that recognize this language. If we are presented with the input string X and both M1 AND M2 accept X we know that it is under intersection.

Complement: TM-recognizable languages are not closed under complement, because recognizers don't always halt.

Problem 4

Show that every infinite TM-recognizable language has an infinite decibadle subset.

Let L be an infinite turning-recognizable language. $\mathcal{E}$ is the enumerator that prints all and only string in L. Since $\mathcal{E}$ never halts if L is infinite, the process will look like this:

1) Run through $\mathcal{E}$. When first string is printed print string x, and set variable temp = x.
2) Continue running through $\mathcal{E}$. When ready to print a new string x2, check to see if $|x_2| > |temp|$. If it is, print x_2 and set temp = x_2. Else, don't print x2
3) Loop back to step 2.

- From these steps, its clear that $\mathcal{E}$ will produce an infinite subset since L itself is infinite. Also, because $\mathcal{E}$ will only print strings in order of length, its language is decidable, so the subset it creates is an infinit decidable subset.