River Kelly
CSCI 338
Test 2
April 5 2021

Question 1

To prove the R1 is uncountable, we will use diagonalization to show, by contradiction, that no correspondence exists between N and R1.

First, let us assume that there does exist a correspondence F between N and R1. Then, F must pair all the members of N with all the members of R1. We must show that F fails to work as it should by finding an x in R1 that is not paired with anything in N , a contradiction.

Assuming that R1 is countable, the elements of R1 may be ordered as r1 , r2 , r3 , . . . rn, and a correspondence F exists between N and R1, the following table shows a few values of a hypothetical correspondence.

n	F(n)
1	0.1111...
2	0.2111...
3	0.3111...
4	0.4111...
...	...

Now, we construct x by taking the elements of the diagonal and complimenting them to ensure that x /= F(n) for and n. Continuing this way, down the diagonal, we obtain all the values of x.

Note, the underlined values in the table above. Also, add 1 value to the diagonal, if 9 then go to 0.

X = 0.2222

But x exists in R1, so then some rn = x. But by the construction of x is different from rn at th n^th spot. A contradiction.

Therefore, R1 is not countable.

Question 2

To show that the above grammar is ambiguous, we must show that there are two or more different leftmost derivations for some string.

S → aSbS|bSaS|e.

There are 3 choices from S.

1. aSbS - Assuming both values for S result in an empty string, the minimal result here must be "ab"
2. bSaS - Assuming both values for S result in an empty string, the minimal result here must be "ba"
3. E - An empty string

Because either of the first two options, 1 and 2, requires that a nonterminal of the alternate type exists following S (i.e. aSb -> ab and bSa -> ba in 1 and 2 respectively), there does not exist a method for which a single string may have 2 distinctly different derivations. The string, at bare minimum, must alternate between a and b (or vice versa).

By definition, this would conclude that the grammar is unambiguous.

Question 3

To determine if L is context free, we can apply the pumping lemma.

FROM THE BOOK

THEOREM 1.70 ....

Pumping lemma If A is a regular language, then there is a number p (the pumping length) where if s is any string in A of length at least p, then s may be divided into three pieces, $s = xyz$, satisfying the following conditions:

1. for each $i \geq 0$, $xy^iz \in A$,
2. $|y| > 0$, and
3. $|xy| \leq p$.

Let s = a^n b^(n+1) C^(n+2). Then, we should be able to decompose s into *uvxyz*.
Note, vwx cannot contain both ‘a’s and ‘c’s because |vwx| <= n.

Given |vwx| <= n, |vx| >= 1.

Suppose vwx is a^i. Lets add an “a” to the string. Making the number a’s n+1, thus i = 2. Then, the resulting string would not be in the language.

Lets add a “b” to the string such that vwx is b^i and i<=n and i>=1. This would make the number of b’s equal to the number of c’s, therefore, not in the language.

If i=0 and vwx is c^i and i<=n and i>=1. Then the number of c’s will be less than or equal to the number of b’s, and not in the language.

Suppose i=0 and vwx is b^i c^j and i+j<=n and i+j>=1. The either the number of b’s is less than or equal to the number of a’s, or the number of c’s is greeted that or equal to b’s. Not in the language.

Finally, suppose vwx is a^i b^j and i+j<=n and i+j=1. When i=2, the number of a’s will be grater than then number of b’s, or the number of b’s greater than the number of c’s. Therefore not in the language.

Considering all 5 possibilities, L is not context-free.

Question 4

PCP is decidable using a unary (1) alphabet.

Suppose we are examining dominos, with values on top and bottom. In the case of a unary alphabet, the top and bottom values only differ in the the number of 1's.

A Turning Machine may decide such by:

1. If the top and bottom have the same number, easy match, accept.
2. If all dominos have more 1's on top than on bottom (or vice versa - more 1's on bottom than on top), no possible match, reject.
3. Locate a domino with more 1's on top than on bottom, and one domino with more 1's on bottom than on top. Choosing the first domino such that its top difference is the same as the difference with the second domino difference on bottom, match. Accept.

PCP is decidable using a unary (1) alphabet.

Question 5

By contradiction, ALL_{TM} is undecidable.

Lets assume that ALL_TM is decidable. Assume there is a Turning Machine R that decides ALL_TM.

Suppose ACCEPT_TM reduces to either SOME_TM or NONE_TM. Then ACCEPT_TM reduces to ALL_TM.

Since ACCEPT_TM is undecidable, but R, using ACCEPT_TM, decides ALL_TM.
A contradiction.

Therefore, ALL_TM is undecidable.