ANKARA UNIVERSITY COMPUTER ENGINEERING
2019-2020 COM364 AUTOMATA THEORY
FINAL EXAMINATION

02.06.2020

Important Note 1: Please write your answers clearly and explain your reasoning. Points will be deducted if your final result is correct but how you obtained it is not clear.
Important Note 2: Your solutions MUST be your own work. Plagiarism will NOT be tolerated.
Important Note 3: Please make sure that the pdf of your solutions is readable.

1. [18 points] Give a regular expression for the following language

$$L = \{0^m 1^n | m + n \text{ is even}\}$$

$$(00)^*(11)^* \cup 0(00)^*1(11)^*$$

2. [15 points] The explanation below is a proof for showing that the language of 0^*1^* is not a regular language. Is this proof correct or incorrect? If it is not correct, explain what the mistake is.

> We can show that 0^*1^* is not regular by using proof by contradiction. Assume it is regular. According to the pumping lemma, there has to be a pumping length, so let p represent the pumping length for this language. We choose $s = 0^p1^p$. Obviously, $s \in 0^*1^*$ and $|s| = 2p \geq p$. We should be able to split this s into three parts as $s = xyz$ such that the three conditions below are true:
>
> (i) $|y| > 0$,
> (ii) $|xy| \leq p$, and
> (iii) for each $i \geq 0$, $xy^iz \in 0^*1^*$
>
> Because of the first two conditions, it has to be the case that y is not empty and it contains only 0s. So, when we consider xy^2z for this string, we know that there will be more 0s than 1s. Then, $xy^2z \notin 0^*1^*$ which means that we have a contradiction. Therefore, 0^*1^* cannot be regular. ▪

The proof is not correct. It should be obvious that the language of 0^*1^* is a regular language. The mistake in the proof is near the end when it says that "So, when we consider xy^2z for this string, we know that there will be more 0s than 1s. Then, $xy^2z \notin 0^*1^*$". Yes, the string xy^2z will have more 0s than 1s but this string is still an element in the language 0^*1^*. So, the claim $xy^2z \notin 0^*1^*$ is wrong.

3. Consider the context-free grammar (CFG) given below for the following questions.

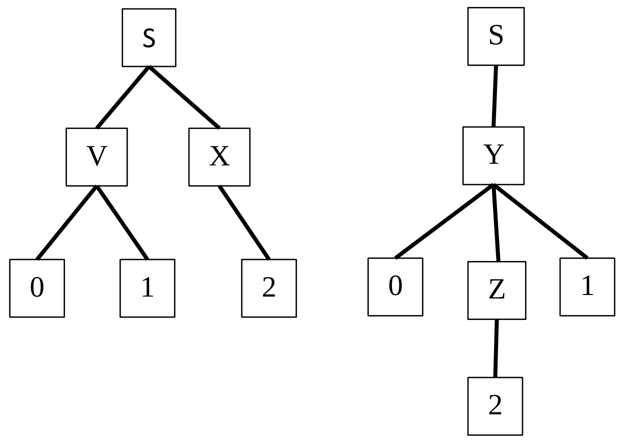
$$S \to VX|Y$$
$$V \to 0V1|01$$
$$X \to X2|2$$
$$Y \to 0Y2|0Z2$$
$$Z \to Z1|1$$

a. [14 points] Describe the language of this grammar at most with one or two sentences. In other words, what kind of strings is generated with this grammar?

$$\{0^i 1^j 2^k \mid i = j \text{ or } i = k\}$$

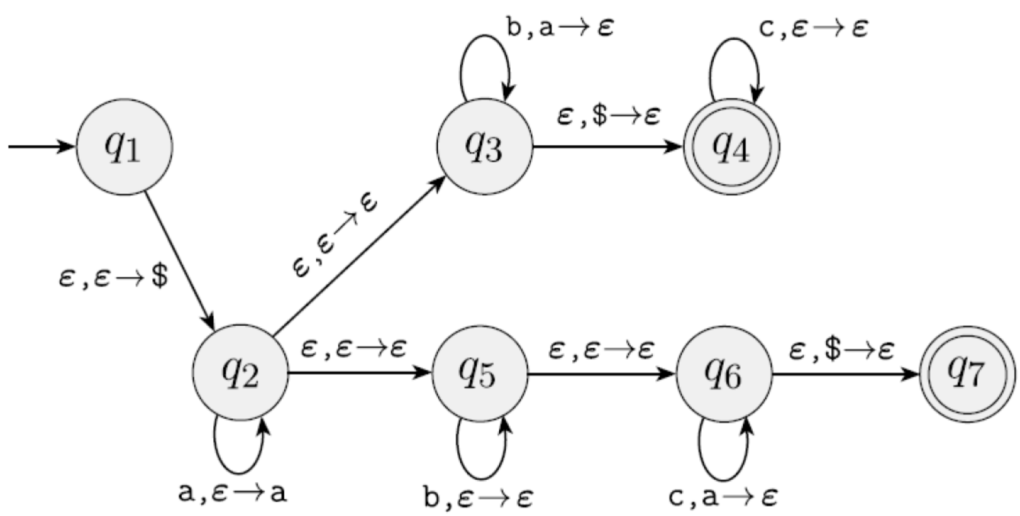
b. [15 points] Is this grammar ambiguous? Why or why not?

Yes, it is ambiguous because it is possible to generate one string with two different parse trees. For example, the string 012 can be obtained with two different parse trees. One in which we first expand the initial S as VX, and another one where S is substituted with Y.



c. [20 points] Give the PDA which recognizes the language of this grammar.

This is actually the same language as one of the examples we discussed in the course. We have the PDA below in the PDA notes. Use 0 instead of a, 1 instead of b, 2 instead of c, and you have the answer for this question.



d. [18 points] Convert the given grammar into an equal one in Chomsky Normal Form (CNF). Show your steps clearly.

Original	Start variable doesn't appear on any of the right sides, so we don't need to introduce a new start variable, but we could. No epsilon rules to remove either. Handle unit rule: S→Y	Then the only problem is introducing variables so that on the right sides we only have either form AB (2 variables) or a (single terminal). Let, A0→0, A1→1, A2→2 and K→A0Y, L→A0Z, M→A0V
$S \rightarrow VX\|Y$ $V \rightarrow 0V1\|01$ $X \rightarrow X2\|2$ $Y \rightarrow 0Y2\|0Z2$ $Z \rightarrow Z1\|1$	$S \rightarrow VX\|0Y2\|0Z2$ $V \rightarrow 0V1\|01$ $X \rightarrow X2\|2$ $Y \rightarrow 0Y2\|0Z2$ $Z \rightarrow Z1\|1$	$S \rightarrow VX\|KA_2\|LA_2$ $V \rightarrow MA_1\|A_0A_1$ $X \rightarrow XA_2\|2$ $Y \rightarrow KA_2\|LA_2$ $Z \rightarrow ZA_1\|1$ $K \rightarrow A_0Y$ $L \rightarrow A_0Z$ $M \rightarrow A_0V$ $A_0 \rightarrow 0$ $A_1 \rightarrow 1$ $A_2 \rightarrow 2$