Student Name: Emily Yeh
Check one:
[X] I completed this assignment without assistance or external resources.
[ ] I completed this assignment with assistance from ___
and/or using these external resources: ___

1A:

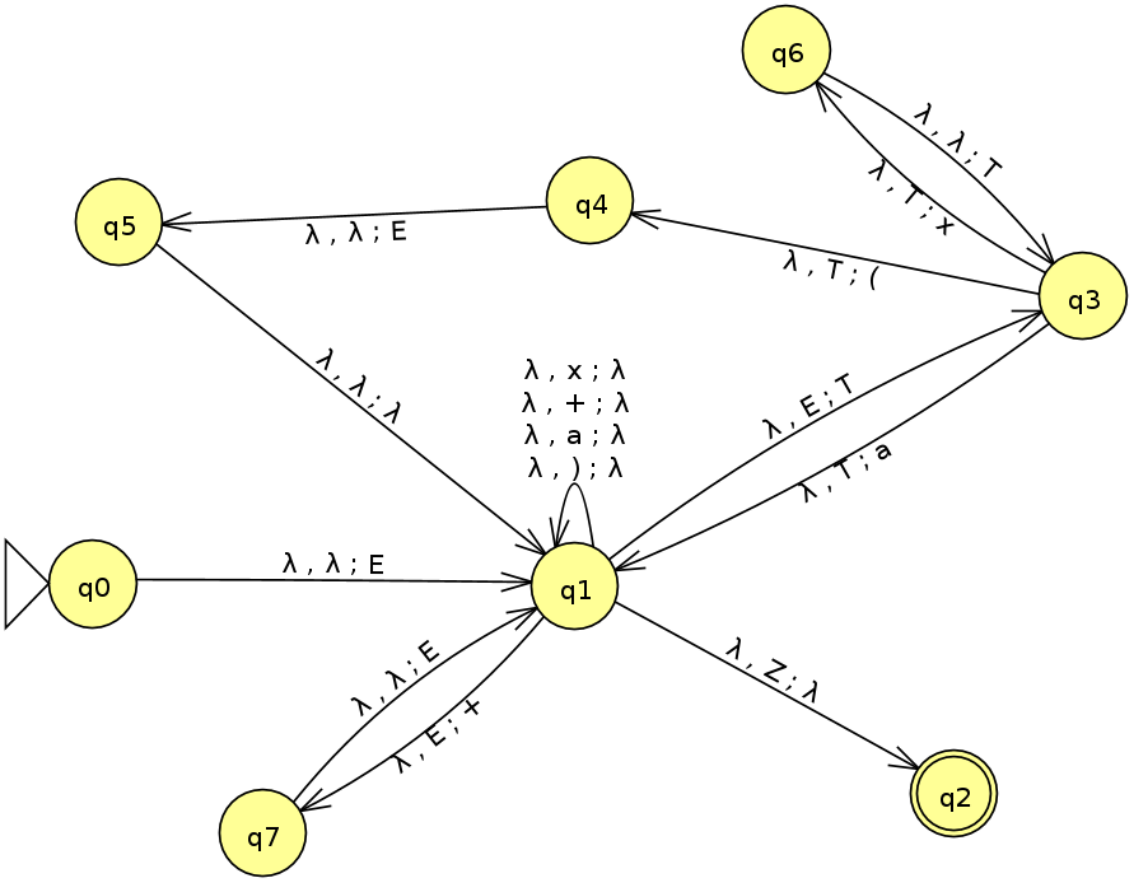
a	**b**	**a XOR (b XOR a)**
F	F	F
F	T	T
T	F	F
T	T	F

1B: In the sentence, “You will eat your meat OR you can’t have any pudding,” it’s implied that…
“You won’t eat your meat AND you can’t have pudding” (F, T) is also true, as is
“You will eat your meat AND you can have pudding.” (T, F) However,
“You won’t eat your meat AND you can have pudding” (F, F) is false, and
“You will eat your meat AND you can’t have pudding” (T, T) is false.
So we have a truth table that looks something like this, where a=”You will eat your meat” and b=”You can’t have any pudding”:

a	**b**	**a ??? b**
T	F	T
F	F	F
T	T	F
F	T	T

This table most closely resembles the a XOR b table (as long as I interpreted the sentence correctly…).

2:



(Sorry it’s so ugly… also, it appears that the default initial value in JFLAP is lambda, so although the first variable being put into the stack from q0, just know that appears to be E, the first item in the stack is actually Z. This definitely confused me at first, so I just wanted to clarify it for whoever is grading this.)

3: In Chomsky Normal Form, there are three acceptable rules:

$A \rightarrow BC$, or

$A \rightarrow a$, or

$S \rightarrow \varepsilon$

Our context-free grammar looks like this:

E -> E + E | T # <- added "| T"

T -> T x T | (E) | a

In CNF, our CFL would look like this:

$E_0 \rightarrow EQ \mid TE_0 \mid \varepsilon$

$T \rightarrow a \mid TR \mid XZ$

$X \rightarrow ($

$Y \rightarrow)$

$Z \rightarrow E)$

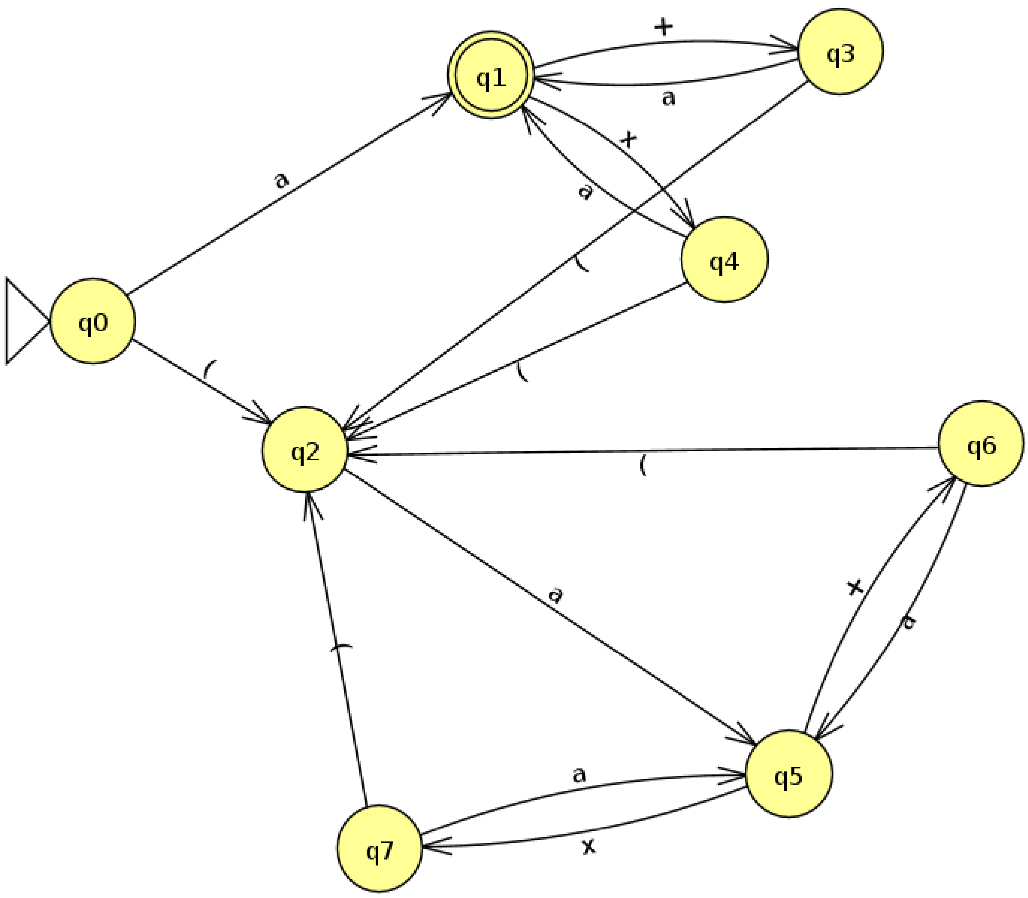
$M \rightarrow x$

$P \rightarrow +$

$Q \rightarrow PE$

$R \rightarrow MT$

4: ~~I think this is a regular language, since it doesn't seem like it's necessary to keep track of the number of each variable (there can be an arbitrary number of Es and Ts).~~ Oh no, just kidding, it's necessary to keep track of the number of left parentheses in order to have the correct number of right parentheses, and since there can be characters between the two, there isn't a way to express this language as a regular expression. I don't think this is a regular language, then, since there's no way to create a regular expression nor a finite state automaton for this. (Although my answer is short, I just spent half an hour trying to draw an FSA in JFLAP, then another 20 minutes trying to derive a regular expression for this grammar. It just doesn't work… right?)



(My failed FSA; I stopped as soon as I realized that it isn't possible to match the parentheses, although I probably should have realized this sooner.)

5A: Let language 1 be L1 = $a^ib^ic^j$ and let language 2 be L2 = $a^ib^jc^j$. If i = 3 and j = 3, then L1 = aaabbbccc and L2 = aaabbbccc, and the intersection of L1 and L2 is aaabbbccc, which is in the language $a^nb^nc^n$. However, if i = 3 and j = 5, then L1 = aaabbbccccc and L2 = aaabbbbbccccc and the intersection of L1 and L2 is aaabbbccccc, which is not in the language $a^nb^nc^n$. Therefore, the set of context-free languages is not closed under intersection.

5B: Let us select a string s = $a^Pb^Pc^P$, where P = the pumping length for $a^nb^nc^n$ that is guaranteed to exist by the pumping lemma. It's obvious that s is a member of $a^nb^nc^n$, but how can we prove that no matter what, s cannot be pumped and therefore, is not context free? Let's divide s into several parts, uvxyz. The rules of the pumping lemma are…

1. For each i >= 0, uv^ixy^iz is a member of the language.
2. |vy| > 0.
3. |vxy| <= p.

From condition 2, we know that v and y must not be empty strings. So what can they contain? According to logic (and Sipser), if v and y contain only one type of symbol, v can't contain both a's and b's or b's and c's, and neither can y. In this case, then, uv^2xy^2z cannot contain equal numbers of a's, b's, and c's, so it can't be a member of our language.

In addition, if v or y contain more than one type of symbol, uv^2xy^2z may contain equal numbers of a's, b's, and c's, but it won't contain them in order, so it can't be a member of our language.

6A:

S -> NP VP
NP -> NP PP
NP -> DET N
VP -> V NP
VP -> VP PP
DET -> a | the
N -> boy | girl | flowers | binoculars
V -> touches | sees
PP -> P NP
P -> in | from | with

"The girl touches the boy with the flower" = DET N V DET N P DET N = NP V NP P NP = NP VP PP or NP V NP P NP

First derivation:

S → NP VP → DET N VP → **DET N VP PP** → DET N V NP PP → DET N V DET N PP → DET N V DET N P NP → DET N V DET N P DET N → the girl touches the boy with the flower

Second derivation:

S → NP VP → DET N VP→ **DET N V NP** → DET N V NP PP → DET N V DET N PP → DET N V DET N P NP → DET N V DET N P DET N → the girl touches the boy with the flower

(I **bolded** the part that is different between the two derivations.)

6B: This sentence either means that the girl touches the boy using the flower, or that the girl touches the boy holding (owning, wearing, etc.) the flower.

6C: The boy sees the girl with the binoculars (DET N V DET N P DET N, same as previous) can mean that either the boy uses binoculars to see the girl, or that the girl has binoculars and the boy sees her.

6D: A→ tall | purple
Make NP→ DET N | DET AN
...And I think this should be it. Let’s do an example: the girl saw the tall boy
S→ NP VP→ DET N VP→ DET N V NP→ DET N V DET AN→ the girl saw the tall boy