Middle East Technical University Department of Computer Engineering

Formal Languages and Abstract Machines
Take Home Exam 2

Koray Can YURTSEVEN
2099547

1 Context-Free Grammars (10 pts)

a) Give the rules of the Context-Free Grammars to recognize strings in the given languages where $\Sigma = \{a, b\}$ and S is the start symbol.

$L(G) = \{w \mid \; w \in \Sigma^*; \; |w| \geq 3;$ (2/10 pts)
the first and the second from the last symbols of w are the same$\}$

Our language has either

$$a(a \cup b)^* a(a \cup b)$$

or

$$b(a \cup b)^* b(a \cup b)$$

To represent this in CFL, we can give the following rules to recognize the language:

$$\begin{aligned} S &\rightarrow aXaY|bXbY \\ X &\rightarrow aX|bX|\varepsilon \\ Y &\rightarrow a|b \end{aligned}$$

$L(G) = \{w \mid \; w \in \Sigma^*; \;$ the length of w is odd$\}$ (2/10 pts)

Our language has

$$(a \cup b)((a \cup b)(a \cup b))^*$$

To represent this in CFL, we can give the following rules to recognize the language:

$$S \to X|Y$$
$$X \to a|b$$
$$Y \to XXY|\varepsilon$$

$L(G) = \{w \mid \ w \in \Sigma^*;\ n(w, a) = 2 \cdot n(w, b)\}$ where $n(w, x)$ is the number of x symbols in w (3/10 pts)

Our language has

$$((aab) \cup (aba) \cup (baa))^*$$

To represent this in CFL, we can give the following rules to recognize the language:

$$S \to SX|\varepsilon$$
$$X \to aab|aba|baa$$

---

b) Find the set of strings recognized by the CFG rules given below: (3/10 pts)

$S \to X \mid Y$
$X \to aXb \mid A \mid B$
$A \to aA \mid a$
$B \to Bb \mid b$
$Y \to CbaC$
$C \to CC \mid a \mid b \mid \varepsilon$

X part of S means that, we have equal number of a's and b's, and in the middle of these numbers, we have either 1 or more a's, or we have 1 or more b's. That is:

$$[a^k(aa^* \cup bb^*)b^k],\ where\ k \geq 0$$

Y parf of S means that, first, we have a number of a's or b's, then we have ba, and at last, we have again a number of a's or b's. That is:

$$[(a \cup b)^*ba(a \cup b)^*]$$

By combining these two result, we have:

$$[a^k(aa^* \cup bb^*)b^k] \cup [(a \cup b)^*ba(a \cup b)^*],\ where\ k \geq 0$$

---

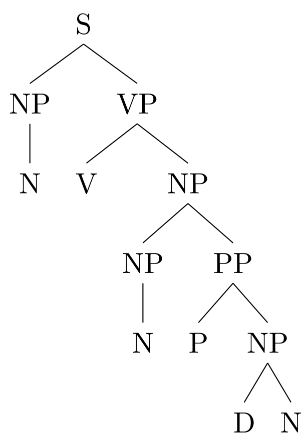
2 Parse Trees and Derivations (20 pts)

Given the CFG below, provide parse trees for given sentences in **a** and **b**.

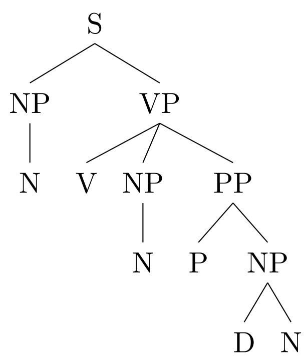
```
S   → NP VP
VP  → V NP | V NP PP
PP  → P NP
NP  → N | D N | NP PP
V   → wrote | built | constructed
D   → a | an | the | my
N   → John | Mary | Jane | man | book | automata | pen | class
P   → in | on | by | with
```

a) Jane constructed automata with a pen (4/20 pts)

Jane is N, constructed is V, automata is N, with is P, a is D, pen is N. Thus we need to have $NVNPDN$ from left to right to have this word. First tree is:



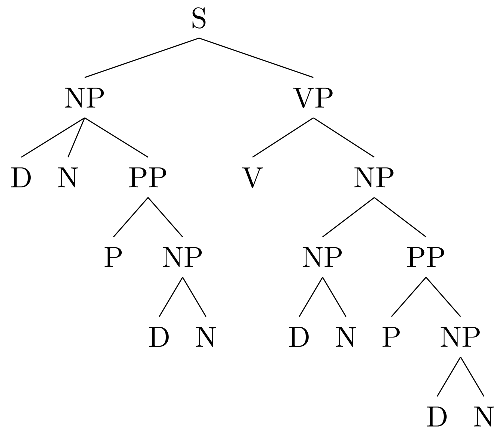
Second tree is:



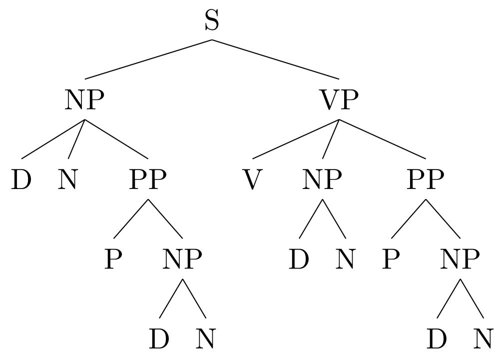
---

b) my book in the man built a Jane by a pen (4/20 pts)

My is D, book is N, in is P, the is D, man is N, built is V, a is D, Jane is N, by is P, a is P, pen is N. Thus we need to have $DNPDNVDNPDN$ from left to right to have this word. First tree is:



Second tree is:



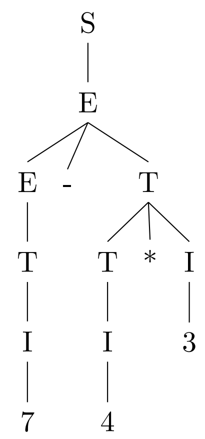
Given the CFG below, answer **c**, **d** and **e**

```
S  → E
E  → E + T | E - T | T
T  → T * I | T / I | I
I  → 0 | 1 | 2 | 3 | 4 | 6 | 7 | 8 | 9
```

c) Provide the left-most derivation of 7 - 4 * 3 step-by-step and plot the final parse tree matching that derivation (4/20 pts)

$$S \to E \to E - T \to T - T \to I - T \to 7 - T \to 7 - T * I \to 7 - I * I \to 7 - 4 * I \to 7 - 4 * 3$$

The tree is:

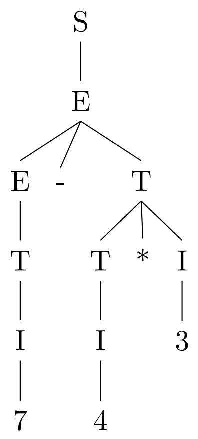


---

d) Provide the right-most derivation of 7 - 4 * 3 step-by-step and plot the final parse tree matching that derivation (4/20 pts)

$$S \to E \to E - T \to E - T * I \to E - T * 3 \to E - I * 3 \to E - 4 * 3 \to T - 4 * 3 \to I - 4 * 3 \to 7 - 4 * 3$$

The tree is:

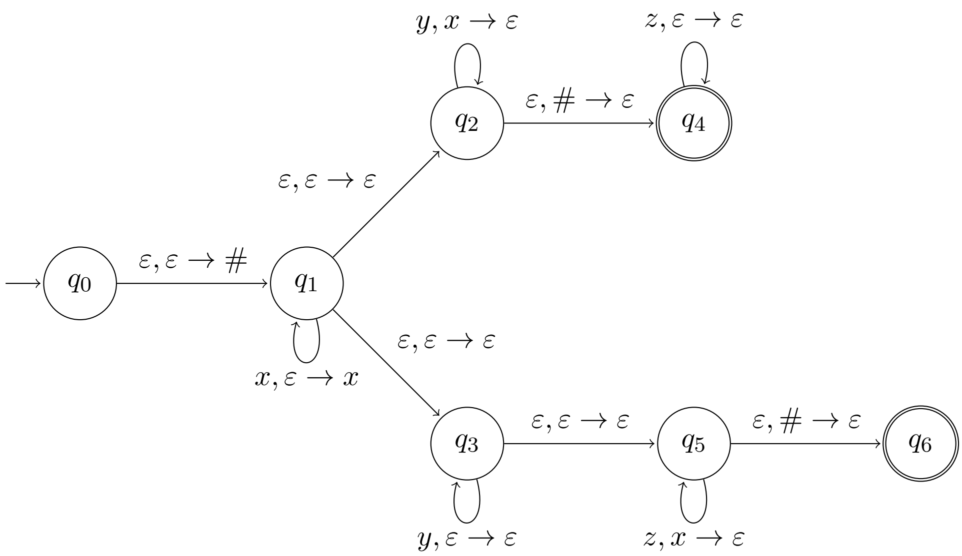


e) Are the derivations in **c** and **d** in the same similarity class? (4/20 pts)

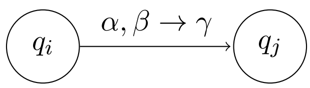
Because they represent applications of the same rules at the same poisitons in the string, only differing the relative order of the applications, they are in the same similarity class. And they are both captured from the same parse tree.

3 Pushdown Automata (30 pts)

a) Find the language recognized by the PDA given below (5/30 pts)



where the transition $((q_i, \alpha, \beta), (q_j, \gamma))$ is represented as:

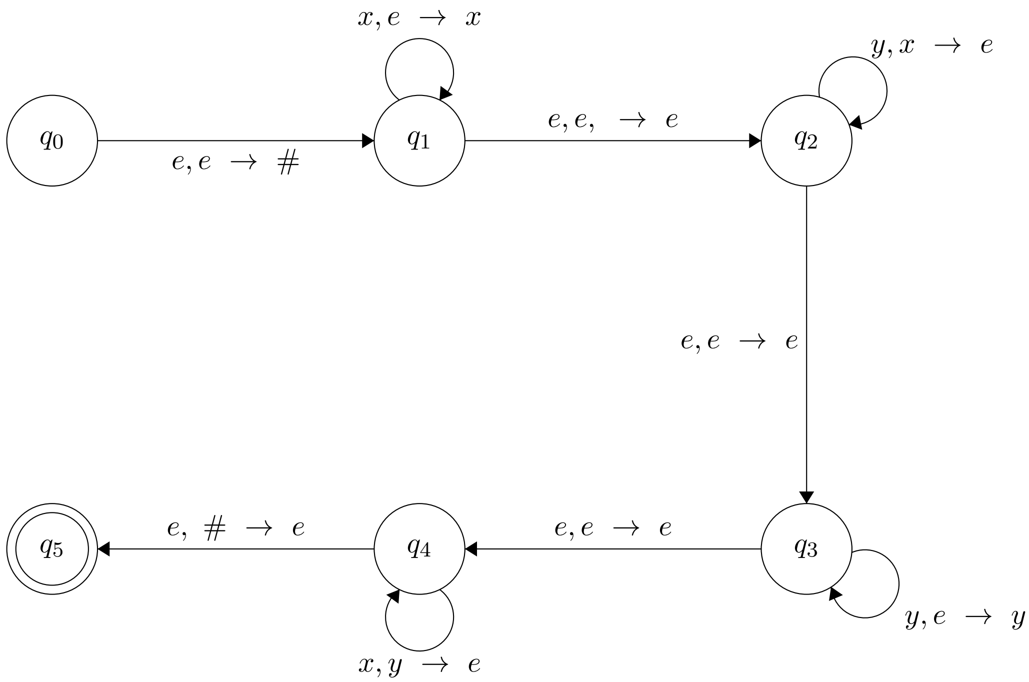


First we denote our stack end as #. Then we read a number of x's and push them on the stack. Let's denote this number k. Now we have two choice, namely:
1) Read k number of y's and while we are reading y's, we are popping the x's. When we have the same number of y's as x's, we will see the bottom of the stack. We move to the q_4 as we are popping the bottom of the stack #. Then we read arbitrary number of z's.
2) Read arbitrary number of y's. Then, we move to q_5 ad start reading z's, and while reading z's we are popping x's. When we see the bottom of the stack, and we have no input to read, we pop # and move to q_6 and finish reading.
Thus, my answer is:

$$L = (x^k y^k z^t) \cup (x^k y^t z^k),\ where\ t, k \geq 0\ ;\ t, k \in N$$

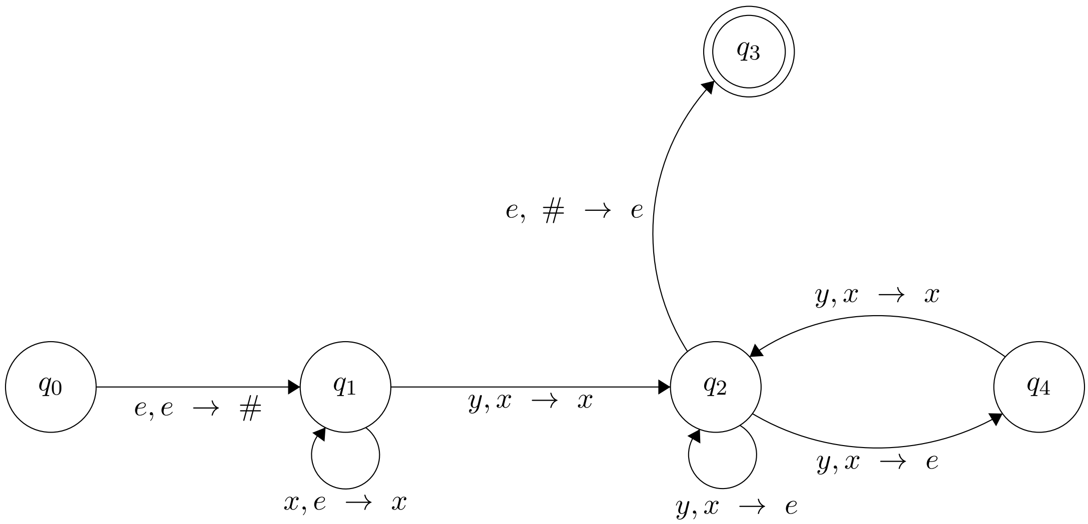
---

b) Design a PDA to recognize language $L = \{x^n y^{m+n} x^m \mid n, m \geq 0;\ n, m \in \mathbb{N}\}$ (5/30 pts)



Initial is q_o
First, mark the stack bottom.
Then, as we read x, push them on the stack.
While reading y's, pop x's that are on the stack.
When we reached to the bottom of the stack, start pushing y's to the stack.
Again, while we are reading x's, pop y's that are on the stack.
When we have no input to read, and we have stack's bottom on the top of the stack, we can finish reading.

c) Design a PDA to recognize language $L = \{x^n y^m \mid n < m \leq 2n;\ n, m \in \mathbb{N}^+\}$ (10/30 pts)
Do not use multi-symbol push/pop operations in your transitions.
Simulate the PDA on strings xxy (with only one rejecting derivation) and $xxyyyy$ (accepting derivation) with transition tables.



Initial state is q_0.
First, we mark the bottom of the stack.
Then, we are pushing x's to the stack.
When we encounter with a y, we don't immediately pop a x. We move to q_2.
When we are reading y's, we either pop an x and stay at q_2 or we move to q_4 while popping a x.
If we are at q_4, we are checking the stack to make our move, because if we don't have a x on the stack, that means we have 2 times more y's than x's.
While at q_2, if we don't have any input to read and our top of the stack is #, we move to q_3 and terminate the reading process.
The simulation for xxy is:

State	Unread Input	Stack
q0	xxy	e
q1	xxy	#
q1	xy	x#
q1	y	xx#
q2	e	xx#

Rejected.
The simulation for $xxyyyy$ is:

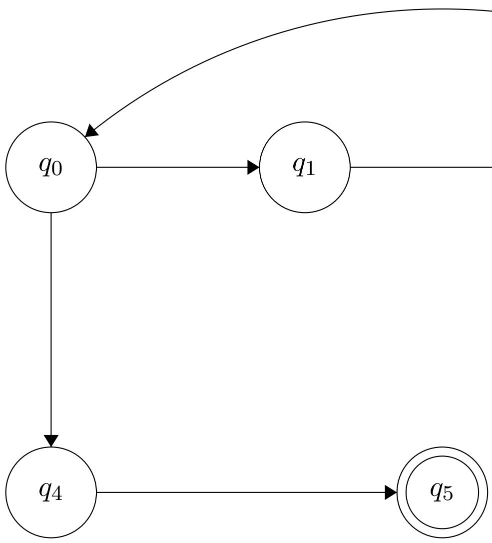
State	Unread Input	Stack
q0	xxyyyy	e
q1	xxyyyy	#
q1	xyyyy	x#
q1	yyyy	xx#
q2	yyy	xx#
q4	yy	x#
q2	y	x#
q2	e	#
q3		

Accepted.

---

d) Given two languages L' and L as $L' = \{w \mid \ w \in L;\ |w| = 4n + 2\ for\ n \in \mathbb{N}\}$ (10/30 pts)
If L is a CFL, show that L' is also a CFL by constructing an automaton for L' in terms of another automaton that recognizes L.

Assume that the construction of such L is given. Also, we can draw an automata for L_2 such that it accepts the same language as L' accepts.



Where q_0 is initial and there is a input reading between those states.
Since L_2 is a regular language, because I have drawn the automata for that, using the Theorem 3.5.2 in the book, the intersection of a CFL and Regular Language is a CFL, we can create the following:

$$L' = L \cap L_2$$

such that, we have a stack to keep tabs on the stack of L', the initial and final states of this language are from both L' and L_2's accepting states, and the states of this language are pair of L' and L_2's states.

---

4 Closure Properties (20 pts)

Let L_1 and L_2 be context-free languages which are not regular, and let L_3 be a regular language. Determine whether the following languages are necessarily CFLs or not. If they need to be context-free, explain your reasoning. If not, give one example where the language is a CFL and a counter example where the language is not a CFL.

a) $L_4 = L_1 \cap (L_2 \setminus L_3)$ (10/20 pts)

Let L_2 and L_3 denotes the following:

$$L_2 = (a^n b^n c^m d^m),\ where\ m, n \geq 1$$
$$L_3 = (a \cup b)^*$$

Thus, L_2 is context free but not regular, and L_3 is a regular language, as in stated in the question.

$$L_2 \setminus L_3 = L_2$$

And the intersection of two context free language is unknown because context free languages are not closed under intersection. Hence, we cannot know whether the L_4 context free or not.

---

b) $L_5 = (L_1 \cap L_3)*$ (10/20 pts)

Let a PDA named P accepts L_1, and let a DFA named M accepts L_3. We can create a new PDA that simulate P and M at the same time on the same input, and accept if both of them accepts. Then this new PDA can accept L_1 and L_3's intersection. Since context free languages are closed under Kleene star operation, L_5 is a context free language. Constructing this new PDA's algorithm is:
The stack of new PDA is the stack of P
The state of this new PDA is te pair of states of P and M
These can determine the transition function of this new PDA
And the final states of this new PDA are P's and M's accepting states
Reference

---

5 Pumping Theorem (20 pts)

a) Show that $L = \{a^n m^n t^i \mid n \leq i \leq 2n\}$ is not a Context Free Language using Pumping Theorem for CFLs. (10/20 pts)

Suppose we have S, where S is in form of uv^nxy^nz and $n \geq 0$.
Then, we should have;
uv^ixy^iz for every $i \geq 0$, $|vy| > 0$, and $|vxy| \leq P$, where P denotes pumping length.
We will have 4 cases, where each is the following:

i) We have at least one t in substring and along with t('s), we have m('s).
ii)We only have at least one t in substring.$\#t \geq 1$
iii)We don't have t's in substring. Then we have either at least one a's or at least one m's in this substring. Not both of them.
iv)We don't have t's in substring. But we have both at least one a's and m's in the substring.

i)Assume the first case: we have at least one t along with m's in this substring vxy.
Then, if we use pump, the number of a's and m's will not be equal, because if we have t, then we don't have any a along with those m's, because the length of substring xvy must be less than our pumping length. Therefore it is a contradiction in this case.

ii)Assume the second case: we have only at least one t.
Then, if we pump $i > P$, the second restriction will not be held ($i \leq 2n$).
For clarification:
we have aaammmttt. If we choose our substring as a t, and if we pump 4 times, which is greater than 3($\#P$), then we will have more than $2n$ t's. If I have choosen to write more than 3 t's in the example, i.e. 4-5-6, then the number of i will be decreased. But $i > P$ holds for every P regardless of the input. Therefore it is a cntradiction in this case.

iii)Assume the third case: We don't have t's in the substring. We have a's or m's, but not both. Then, it is easy to see that if we pump one of them, the equation of their length will not be true. Therefore we cannot accept this state either.

iv)Assume the forth and the last case: We don't have t's in the substring, but we have a's and m's together.
As in stated in the second case, if we pump this substring more than P time, $i > P$, the second restriction will not be held.
For clarification:
We have aaammmtttttt. We choose our substring as am. Our pumping length is 3. If we pump this substring 4 times, more than 3 times, we will have more than 7 a's and b's, but we have at most 6 t's at the first step.

Since all the cases above are contradiction, this language is not a CFL.

---

b) Show that $L = \{a^n b^{2n} a^n \mid n \in \mathbb{N}+\}$ is not a Context Free Language using Pumping Theorem for CFLs. (10/20 pts)

Suppose we have S, where S is in form of $uv^n xy^n z$ and $n \geq 0$.
Then, we should have;
$uv^i xy^i z$ for every $i \geq 0$, $|vy| > 0$, and $|vxy| \leq P$, where P denotes pumping length.
Assume that we have only a's in this substring vxy. Then with any $i \geq 0$, we have an imbalance between a's and b's. Therefore our substring cannot contain only a's.
Which brings us to the second case, where our substring contains both a's and b's. To avoid imbalance, we should have a number of a's, then a number of b's and at last, a number of a's again. However, since our substring is continous, it should contain all b's in order to connect both a's in different sides. i.e. oursubstring is in this form:

$$aaa...aaabbb...bbbbaaa....a$$

In this form, we don't have any relation of number of a's in the left of b's and the number of a's in the right of the b's. Also we do not have the rule for number of a's is half of the number of b's for each side. One can see that, to obtain such substring, we need to have all b's. Since our pumping length is P, and the number of b's is $2P$, we have a contradiction in $|vxy| \leq P$, therefore we cannot have this kind of string either.
The third case is, our substring can contain only b's. It is easy to see that it will cause an imbalance between the number of a's and the number of b's, because when i increases, only the number of b's increases, and we won't have the equality for each i.
All in all, all cases that don't contain all b's break symmetry, and the case that contains all b's is not applicable, because it contradicts with the pumping lemma. Therefore, we cannot have this string in our language.

---

6 CNF and CYK (not graded)

a) Convert the given context-free grammar to Chomsky Normal Form.

$S \to XSX \mid xY$
$X \to Y \mid S$
$Y \to z \mid \varepsilon$

answer here ...

b) Use the grammar below to parse the given sentence using Cocke–Younger–Kasami algorithm. Plot the parse trees.

S → NP VP
S → X1 VP
X1 → Aux NP
S → book | include | prefer
S → Verb NP
S → X2 PP
S → Verb PP
S → VP PP
NP → I | she | me | Houston
NP → Det Nom
Nom → book | flight | meal | money
Nom → Nom Noun
Nom → Nom PP

VP → book | include | prefer
VP → Verb NP
VP → X2 PP
X2 → Verb NP
VP → Verb PP
VP → VP PP
PP → Prep NP
Det → that | this | the | a
Noun → book | flight | meal | money
Verb → book | include | prefer
Aux → does
Prep → from | to | on | near | through

book the flight through Houston

Empty parse table:

1:5 → 1:1 2:5 1:5 → 1:2 3:5 1:5 → 1:3 4:5 1:5 → 1:4 5:5				
1:4 → 1:1 2:4 1:4 → 1:2 3:4 1:4 → 1:3 4:4	2:5 → 2:2 3:5 2:5 → 2:3 4:5 2:5 → 2:4 5:5			
1:3 → 1:1 2:3 1:3 → 1:2 3:3	2:4 → 2:2 3:4 2:4 → 2:3 4:4	3:5 → 3:3 4:5 3:5 → 3:4 5:5		
1:2 → 1:1 2:2	2:3 → 2:2 3:3	3:4 → 3:3 4:4	4:5 → 4:4 5:5	
1:1	2:2	3:3	4:4	5:5
book	the	flight	through	Houston

rest of the answer here ...

7 Deterministic Pushdown Automata (not graded)

Provide a DPDA to recognize the given languages, the DPDA must read its entire input and finish with an empty stack.

a) $a^*bc \cup a^nb^nc$

answer here ...

b) $(aa)^*c \cup a^nb^nc$

answer here ...