Middle East Technical University Department of Computer Engineering

Formal Languages and Abstract Machines
Take Home Exam 2

Yavuz Selim YESILYURT
2259166

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$\}$

S → aXaA | bXbA
X → aX | bX | e
A → a | b

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

S → aX | bX
X → aS | bS | e

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

S → aaSbS | bSaaS | abSaS | baSaS | aSabS | aSbaS | e

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

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

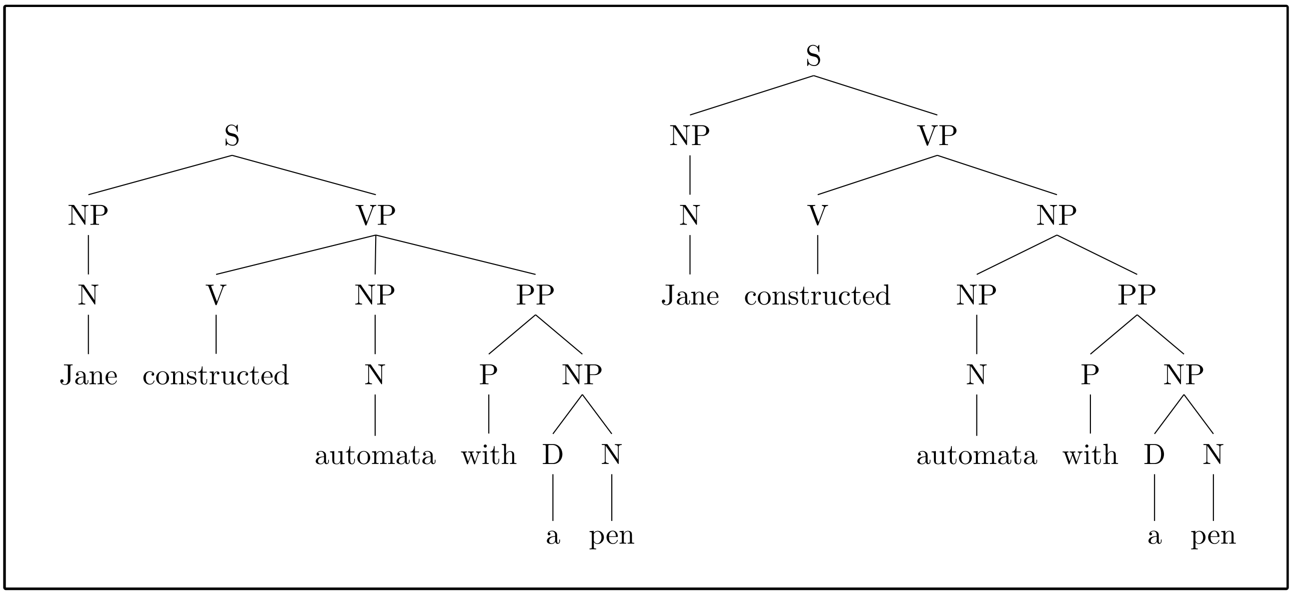
$$L(G) = \{w \mid\ w \in \{a,b\}^*;\ contains\ ba\ or\ in\ the\ form\ a^x b^y\ where\ x \neq y,\ x,y \geq 1\}$$

2 Parse Trees and Derivations (20 pts)

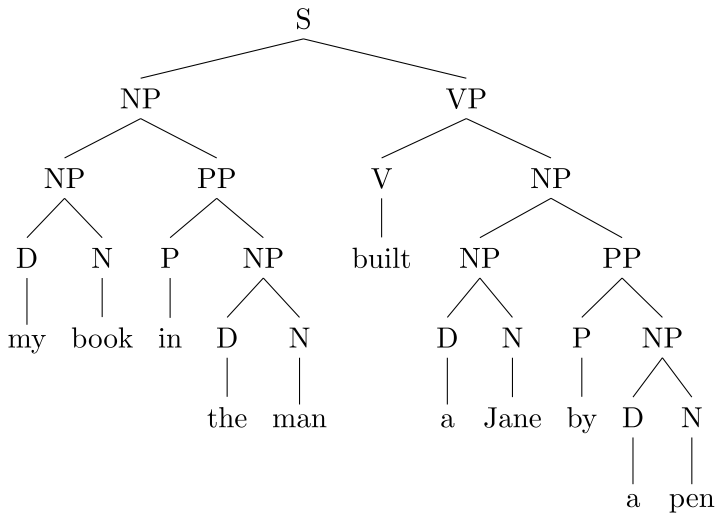
Given the CFG below, provide parse trees for given sentences in **a** and **b**.

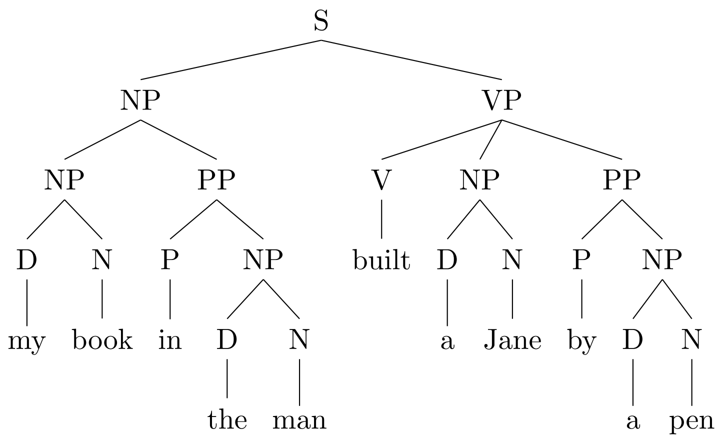
```
S   → NP VP
VP  → V NP | V NP PP
PP  → P NP
NP  → N | D N | NP PP
V   → wrote | built | constructed
D   → a | an | the | my
N   → John | Mary | Jane | man | book | automata | pen | class
P   → in | on | by | with
```

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



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





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

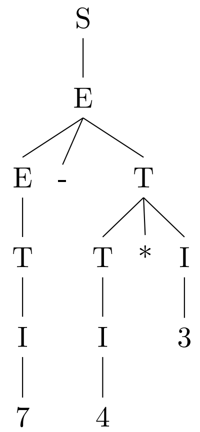
```
S  → E
E  → E + T | E - T | T
T  → T * I | T / I | I
I  → 0 | 1 | 2 | 3 | 4 | 6 | 7 | 8 | 9
```

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)

Left-most derivation:

S ⇒ E ⇒ E - T ⇒ T - T ⇒ I - T ⇒ 7 - T ⇒ 7 - T * I ⇒ 7 - I * I ⇒ 7 - 4 * I ⇒ 7 - 4 * 3

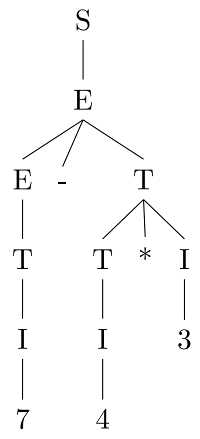
Corresponding parse tree:



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)

Right-most derivation:
$S \Rightarrow E \Rightarrow E - T \Rightarrow E - T * I \Rightarrow E - T * 3 \Rightarrow E - I * 3 \Rightarrow E - 4 * 3 \Rightarrow T - 4 * 3 \Rightarrow I - 4 * 3 \Rightarrow 7 - 4 * 3$

Corresponding parse tree:

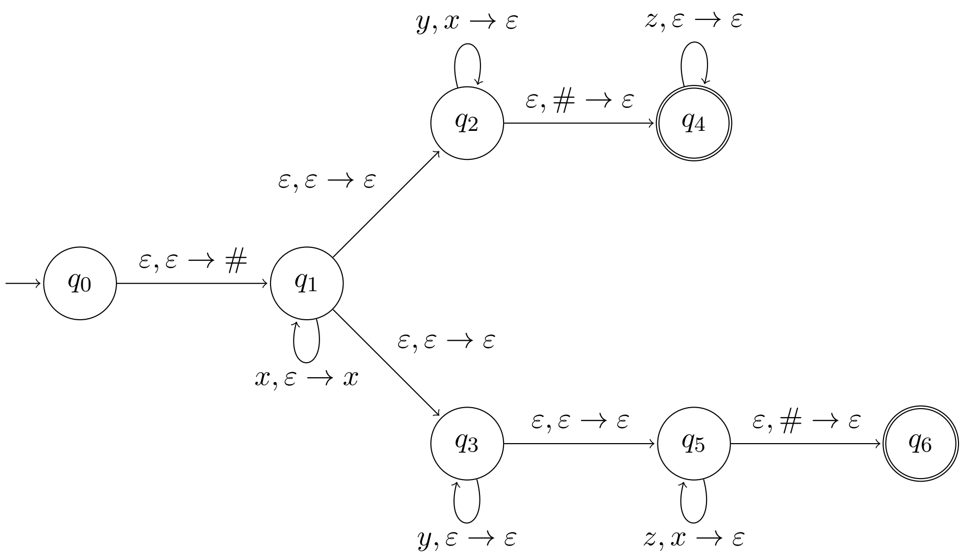


e) Are the derivations in **c** and **d** in the same similarity class? (4/20 pts)

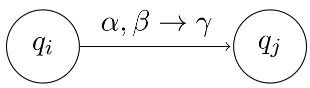
Yes, because from the definition, we say, two derivations D and D' are similar if the pair (D, D') belongs in the reflexive, symmetric, transitive closure of $\prec$. In other words two derivations are similar if they can be transformed into another via a sequence of "switchings" in the order in which rules are applied. Such a "switching" can replace a derivation either by one that precedes it, or by one that it precedes. In our example we found a left-most derivation and a right-most derivation for a word with the rules given. As can be seen clearly, we can reach to the right-most derivation via a sequence of "switchings" on left-most derivation and vice-versa applies. Besides two derivations gave us the same parse trees, which implies that these derivations are in the same similarity class.

3 Pushdown Automata (30 pts)

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

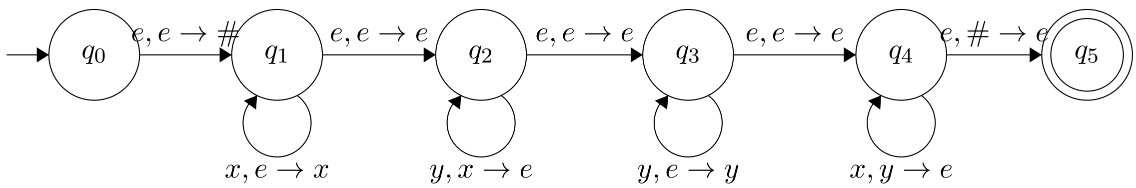


where the transition $((q_i, \alpha, \beta), (q_j, \gamma))$ is represented as:



Language recognized by the PDA above is:
$L = \{x^n y^n z^m \ or \ x^n y^m z^n \mid n, m \geq 0; \ n, m \in \mathbb{N}\}$

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)

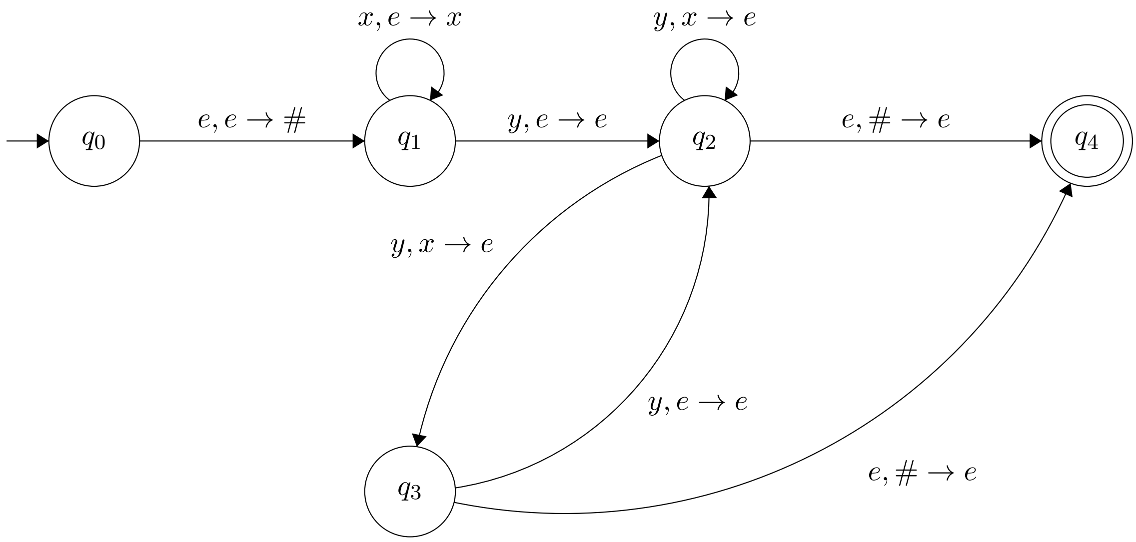


c) Design a PDA to recognize language $L = \{x^n y^m \mid n < m \leq 2n;\ n, m \in \mathbb{N}^+\}$ (10/30 pts)
Do not use multi-symbol push/pop operations in your transitions.
Simulate the PDA on strings xxy (with only one rejecting derivation) and $xxyyyy$ (accepting derivation) with transition tables.

Let $M = (K, \Sigma, \Gamma, \Delta, s, F)$, where $K = \{q_0, q_1, q_2, q_3, q_4\}$, $\Sigma = \{x, y\}$, $\Gamma = \{x, y\}$, $s = q_0$, $F = \{q_4\}$ and Δ contains eight transitions;

$$
\begin{array}{c}
1.((q_0, e, e), (q_1, \#)) \\
2.((q_1, x, e), (q_1, x)) \\
3.((q_1, y, e), (q_2, e)) \\
4.((q_2, y, x), (q_2, e)) \\
5.((q_2, y, x), (q_3, e)) \\
6.((q_2, e, \#), (q_4, e)) \\
7.((q_3, y, e), (q_2, e)) \\
8.((q_3, e, \#), (q_4, e))
\end{array}
$$

The PDA figure is given below;



For the xxy simulation;

State	*Unread Input*	*Stack*	*Transition Used*
q_0	xxy	e	-
q_1	xxy	$\#$	1
q_1	xy	$x\#$	2
q_1	y	$xx\#$	2
q_2	e	$xx\#$	3

As can be seen in the last row of the transition table, input word ended but stack is not empty and the machine is not in the final state, since there is not a suitable transition for that case in our PDA the word is not accepted by the Language.

For the *xxyyyy* simulation;

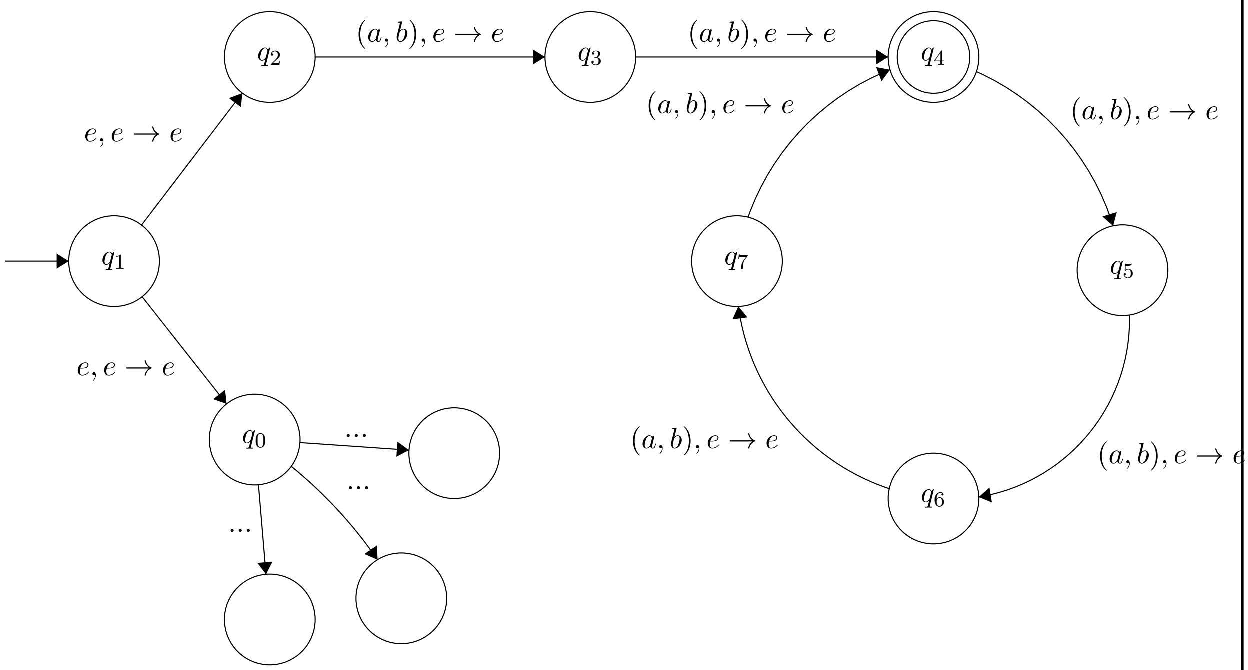
State	*Unread Input*	*Stack*	*Transition Used*
q_0	$xxyyyy$	e	-
q_1	$xxyyyy$	$\#$	1
q_1	$xyyyy$	$x\#$	2
q_1	$yyyy$	$xx\#$	2
q_2	yyy	$xx\#$	3
q_3	yy	$x\#$	5
q_2	y	$x\#$	7
q_3	e	$\#$	5
q_4	e	e	8

As can be seen in the last row of the transition table, input word and stack contents are empty and the machine is in final state, therefore word is accepted by the language.

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.

If L is a CFL then it has a push down automata that recognizes L with its alphabets, transitions and states. To construct an automaton for L' in terms of another automaton that recognizes L we will make some additions on the CFL of L and get the final automata. So lets say L has a PDA $M = (K, \Sigma, \Gamma, \Delta, s, F)$, where K contains some states including its initial state and final state(s), let $s = q_0$ and Δ contains some transitions.

To construct the automata that recognizes both languages we will design a PDA for L' and then bind it with the PDA of L with adding a new state and adding transitions to it for both languages, namely we will create a new initial state for final automata and add two transitions to it to combine these PDA'S. So let us illustrate that;



q_1 is the initial state of the automata that recognizes both languages and as can be seen from the figure above it has two e transitions to q_2 and q_0 representing automata that recognizes L' and automata that recognizes L respectively. Since we do not know the contents of the language L, I just added an e transition from the initial state q_1 to its older initial state q_0 and I left the other transitions and states related with L blank, which means they can be in any form that will construct the PDA for recognizing L.

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)

$L_2 - L_3$ also equals to $L_2 \cap L_3'$, since L_3 is a regular language, its closed under complementation so L_3' is also a regular language. We also know that the intersection of a Context-free language and a regular language is Context-free, therefore we can say $L_2 \cap L_3'$ (also $L_2 - L_3$) is context-free. We know the intersection of two CFL is not necessarily Context-free, so since L_1 and the righthand-side are Context free we can not say L_4 is necessarily context-free. For example in the case of $L_1 = \{a^n b^n c^m | m, n \geq 0\}$ and rhs equals to $\{a^n b^m c^m | m, n \geq 0\}$, $L_4 = \{a^n b^n c^n | n \geq 0\}$ which is not a Context-free language, but for example in the case of $L_1 = \{a^n b^n | n \geq 0\}$ and rhs equals to $\{a^{2n} b^{2n} | n \geq 0\}$, $L_4 = \{a^n b^n | n \geq 0\}$ which is a Context-free language.

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

L_1 is a CFL and L_3 is a Regular Language, since we know that the intersection of a CFL and a regular language is context-free, we can easily say that intersection of L_1 and L_3 is context-free and since the CFL are closed under Kleene Star operation, the Kleene star of the expression is also context-free, therefore L_5 is context-free.

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 (10/20 pts)
using Pumping Theorem for CFLs.

Assume that L is a CFL, Then by Pumping Lemma there exists a pumping length k, depending on the grammar, such that any word $w \in L$ of length greater than k can be re-written as $w = uvxyz$ for $v \neq e$ or $y \neq e$. We will apply the strong version of the lemma which additionally says $|vxy| < k$. Let our $w = a^k m^k t^{2k}$ and we have $|vy| \geq 1$ and $|vxy| < k$. Depending on the where we pump vxy to w there are 5 cases.

Cases 1 and 2: We can either pump vxy for $i = 1$ within the a's on the first part of the word (vxy only consists of a's) or within the m's on the second part of the word (vxy only consists of m's) , and in each case we get: $w' = a^x m^y t^{2k}$ (in case 1 $x = k + 1,\ y = k$, in case 2 $x = k,\ y = k + 1$) where $x \neq y$, which constitutes a contradiction.

For case 3: We can pump vxy to the third part of the word which consists of only t's. In this case, when we pump vxy with $i = 1$, we get $w' = a^k m^k t^x$, where $x > 2k$ namely $x = 2k + 1$ which contradicts with the form of the word.

For case 4: We can pump vxy in between first and second part of the word, namely it can contain some a's and some m's. In this case if we pump vxy for $i = 0$ we get, $w' = a^x m^y t^{2k}$ where either $x \neq y$ or if they are equal $2x < 2k$ which creates a contradiction.

For case 5: We can pump vxy in between second and third part of the word, namely it can contain some m's and some t's. In this case if we pump vxy for $i = 0$ we get, $w' = a^k m^x t^y$ where either $k \neq x$ or if they are equal $y < 2k$ which creates a contradiction.

In each case we reached to a contradiction, therefore we can say L is not Context Free Language.

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)

Assume that L is a CFL, Then by Pumping Lemma there exists a pumping length k, depending on the grammar, such that any word $w \in L$ of length greater than k can be re-written as $w = uvxyz$ for $v \neq e$ or $y \neq e$. We will apply the strong version of the lemma which additionally says $|vxy| < k$. Let our $w = a^k b^{2k} a^k$ and we have $|vy| \geq 1$ and $|vxy| < k$. Depending on the where we pump vxy to w there are again 5 cases.

Cases 1 and 3: We can either pump vxy for $i = 1$ within the a's on the first part of the word or within the a's on the third part of the word (vxy only consists of a's in both cases), and in each case we get: $w' = a^x m^{2k} t^y$ (in case 1 $x = k+1,\ y = k$, in case 3 $x = k,\ y = k+1$) where $x \neq y$, which constitutes a contradiction.

For case 2: We can pump vxy to the second part of the word which consists of only b's. In this case, when we pump vxy with $i = 1$, we get $w' = a^k b^x a^k$, where $x > 2k$ namely $x = 2k+1$ which contradicts with the form of the word.

For case 4: We can pump vxy in between first and second part of the word, namely it can contain some a's and some b's. In this case if we pump vxy for $i = 0$ we get, $w' = a^x b^y a^k$ where either $x \neq k$ or if they are equal $y < 2k$ which creates a contradiction.

For case 5: We can pump vxy in between second and third part of the word, namely it can contain some b's and some a's. In this case if we pump vxy for $i = 0$ we get, $w' = a^k m^x t^y$ where either $k \neq y$ or if they are equal $x < 2k$ which creates a contradiction.

In each case we reached to a contradiction, therefore we can say L is not Context Free Language.