Vivekananda
College of Engineering & Technology
Nehru Nagar Post, Puttur, D.K. 5742013

Lecture Notes

15CS54
Automata Theory and Computability
(CBCS Scheme)

Prepared by

Mr. Harivinod N
Dept. of Computer Science and Engineering,
Vivekananda College of Engineering and Technology, Puttur

Module-4: Properties of CFL, Turing Machine

Contents

Course website:
www.techjourney.in

1. Where Do the Context-Free Languages Fit in the Big Picture?

Theorem: The regular languages are a proper subset of the context-free languages.

Proof: We first show that every regular language is context-free. We then show that there exists at least one context-free language that is not regular.

We show that every regular language is context-free by construction. If L is regular, then it is accepted by some DFSM $M = (K, \Sigma, \delta, s, A)$. From M we construct a PDA $M' = (K', \Sigma', \Gamma', \Delta', s', A')$ to accept L. In essence, M' will simply be M and will ignore the stack. Let M' be $(K, \Sigma, \varnothing, \Delta', s, A)$, where Δ' is constructed as follows: For every transition (q_i, c, q_j) in δ, add to Δ' the transition $((q_i, c, \varepsilon), (q_j, \varepsilon))$. M' behaves identically to M, so $L(M) = L(M')$. So the regular languages are a subset of the context-free languages.

The regular languages are a *proper* subset of the context-free languages because there exists at least one language, A^nB^n, that is context-free but not regular.

Theorem: There is a countably infinite number of context free languages.

Proof: Every context-free language is generated by some context free grammar G = (V, Σ, R, S). *Upper-bound:* We can encode the elements of V as binary strings, so we can lexicographically enumerate all the syntactically legal context free grammars. There cannot be more context-free languages than there are context-free grammars. So, there is at most a countably infinite number of context-free languages.

There is not a one-to-one relationship between context-free languages and context-free grammars since there is an infinite number of grammars that generate any given language. But every regular language is context free (*lower-bound)*. Also, there is a countably infinite number of regular languages.

So, there is at least and at most a countably infinite number of context-free languages.

2. Closure Theorems for Context-Free Languages

Unfortunately, there are fewer closure theorems than regular languages.

The context-free languages are closed under:

- Union
- Concatenation
- Kleene star
- Reverse
- Letter substitution

Proof: The context-free languages are closed under Union

The context-free languages are closed under union: If L_1 and L_2 are context-free languages, then there exist context-free grammars $G_1 = (V_1, \Sigma_1, R_1, S_1)$ and $G_2 = (V_2, \Sigma_2, R_2, S_2)$ such that $L_1 = L(G_1)$ and $L_2 = L(G_2)$. If necessary, rename the nonterminals of G_1 and G_2 so that the two sets are disjoint and so that neither includes the symbol S. We will build a new grammar G such that $L(G) = L(G_1) \cup L(G_2)$. G will contain all the rules of both G_1 and G_2. We add to G a new start symbol, S, and two new rules, $S \rightarrow S_1$ and $S \rightarrow S_2$. The two new rules allow G to generate a string iff at least one of G_1 or G_2 generates it. So $G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R_1 \cup R_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}, S)$.

Proof: The context-free languages are closed under Concatenation

The context-free languages are closed under concatenation: If L_1 and L_2 are context-free languages, then there exist context-free grammars $G_1 = (V_1, \Sigma_1, R_1, S_1)$ and $G_2 = (V_2, \Sigma_2, R_2, S_2)$ such that $L_1 = L(G_1)$ and $L_2 = L(G_2)$. If necessary, rename the nonterminals of G_1 and G_2 so that the two sets are disjoint and so that neither includes the symbol S. We will build a new grammar G such that $L(G) = L(G_1)\, L(G_2)$. G will contain all the rules of both G_1 and G_2. We add to G a new start symbol, S, and one new rule, $S \rightarrow S_1S_2$. So $G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R_1 \cup R_2 \cup \{S \rightarrow S_1 S_2\}, S)$.

Proof: The context-free languages are closed under Kleene Star

The context-free languages are closed under Kleene star: If L_1 is a context-free language, then there exists a context-free grammar $G_1 = (V_1, \Sigma_1, R_1, S_1)$ such that $L_1 = L(G_1)$. If necessary, rename the nonterminals of G_1 so that V_1 does not include the symbol S. We will build a new grammar G such that $L(G) = L(G_1)^*$. G will contain all the rules of G_1. We add to G a new start symbol, S, and two new rules, $S \rightarrow \varepsilon$ and $S \rightarrow S\,S_1$. So $G = (V_1 \cup \{S\}, \Sigma_1, R_1 \cup \{S \rightarrow \varepsilon, S \rightarrow S\,S_1\}, S)$.

Proof: The context-free languages are closed under Reverse

The context-free languages are closed under reverse: Recall that $L^R = \{w \in \Sigma^* : w = x^R \text{ for some } x \in L\}$. If L is a context-free language, then it is generated by some Chomsky normal form grammar $G = (V, \Sigma, R, S)$. Every rule in G is of the form $X \rightarrow BC$ or $X \rightarrow a$, where X, B, and C are elements of $V - \Sigma$ and $a \in \Sigma$. In the latter case $L(X) = \{a\}$. $\{a\}^R = \{a\}$. In the former case, $L(X) = L(B)L(C)$. By Theorem 2.4, $(L(B)L(C))^R = L(C)^R L(B)^R$. So we construct, from G, a new grammar G', such that $L(G') = L^R$. $G' = (V_G, \Sigma_G, R', S_G)$, where R' is constructed as follows:

- For every rule in G of the form $X \rightarrow BC$, add to R' the rule $X \rightarrow CB$.
- For every rule in G of the form $X \rightarrow a$, add to R' the rule $X \rightarrow a$.

The context-free languages are not closed under intersection, complement or difference.

Proof: The context-free languages are not closed under intersection:

The proof is by counterexample. Let:

$L_1 = \{a^n b^n c^m: n, m \geq 0\}$ /* equal a's and b's. $L_2 = \{a^m b^n c^n: n, m \geq 0\}$ /* equal b's and c's.

Both L_1 and L_2 are context-free, since there exist straightforward context-free grammars for them. But now consider: $L = L_1 \cap L_2 = \{a^n b^n c^n: n \geq 0\}$ which not a CFL. (we prove this using pumping theorem ; discussed in section 3)

Proof: The context-free languages are not closed under complement:

Closure under complement implies closure under intersection, since:

$$L_1 \cap L_2 = \neg(\neg L_1 \cup \neg L_2)$$

The context-free languages are closed under union, so if they were closed under complement, they would be closed under intersection (which they are not).

Proof: The context-free languages are not closed under difference (subtraction):

Given any language L. $\neg L = \Sigma^* - L$

Σ^* is context-free. So, if the context-free languages were closed under difference, the complement of any context-free language would necessarily be context-free. But we just showed that that is not so.

Closure under intersection/difference with the Regular languages

Theorem: The context-free languages are closed under intersection with the regular languages.

Proof: The proof is by construction. If L_1 is context-free, then there exists some PDA $M_1 = (K_1, \Sigma, \Gamma_1, \Delta_1, s_1, A_1)$ that accepts it. If L_2 is regular then there exists a DFSM $M_2 = (K_2, \Sigma, \delta, s_2, A_2)$ that accepts it. We construct a new PDA, M_3 that accepts $L_1 \cap L_2$. M_3 will work by simulating the parallel execution of M_1 and M_2. The states of M_3 will be ordered pairs of states of M_1 and M_2. As each input character is read, M_3 will simulate both M_1 and M_2 moving appropriately to a new state. M_3 will have a single stack, which will be controlled by M_1. The only slightly tricky thing is that M_1 may contain ε-transitions. So M_3 will have to allow M_1 to follow them while M_2 just stays in the same state and waits until the next input symbol is read.

$M_3 = (K_1 \times K_2, \Sigma, \Gamma_1, \Delta_3, (s_1, s_2), A_1 \times A_2)$, where Δ_3 is built as follows:

- For each transition $((q_1, a, \beta), (p_1, \gamma))$ in Δ_1, and each transition $((q_2, a), p_2)$ in δ, add to Δ_3 the transition:
  $$(((q_1, q_2), a, \beta), ((p_1, p_2), \gamma)).$$
- For each transition $((q_1, \varepsilon, \beta), (p_1, \gamma))$ in Δ_1, and each state q_2 in K_2, add to Δ_3 the transition:
  $$(((q_1, q_2), \varepsilon, \beta), ((p_1, q_2), \gamma)).$$

Theorem: The difference $(L_1 - L_2)$ between a context-free language L_1 and a regular language L_2 is context-free.

Proof: $L_1 - L_2 = L_1 \cap \neg L_2$. If L_2 is regular, then, since the regular languages are closed under complement, $\neg L_2$ is also regular. Since L_1 is context-free, by Theorem 13.7, $L_1 \cap \neg L_2$ is context-free.

Using Closure Theorems to Prove a Language Context-Free

Consider the perhaps contrived language $L = \{a^n b^n : n \geq 0 \text{ and } n \neq 1776\}$. Another way to describe L is that it is $\{a^n b^n : n \geq 0\} - \{a^{1776} b^{1776}\}$. $A^n B^n = \{a^n b^n : n \geq 0\}$ is context-free. We have shown both a simple grammar that generates it and a simple PDA that accepts it. $\{a^{1776} b^{1776}\}$ is finite and thus regular. So, by Theorem 13.8, L is context free.

We have so far seen two techniques that can he used to show that language L is context-free:

1. Exhibit a context-free grammar for L.
2. Exhibit a PDA for L.

The third method is

3. Use the closure properties of context-free languages.

3. The Pumping Theorem for Context-Free Languages

Theorem: The length of the yield of any tree T with height h and branching factor b is $\leq b^h$.

Proof: The proof is by induction on h. If h is 1, then just a single rule applies. So the longest yield is of length less than or equal to b. Assume the claim is true for $h = n$. We show that it is true for $h = n + 1$. Consider any tree with $h = n + 1$. It consists of a root, and some number of subtrees, each of which is of height $\leq n$. By the induction hypothesis, the length of the yield of each of those subtrees is $\leq b^n$. The number of subtrees of the root is $\leq b$. So the length of the yield must be $\leq b(b^n) = b^{n+1} = b^h$.

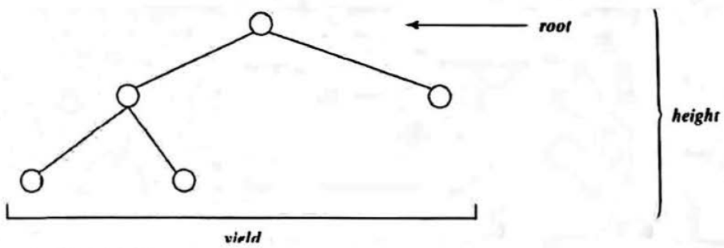


FIGURE 13.1 The structure of a parse tree.

Pumping Theorem

Statement: If L is a context-free language, then $\exists k \geq 1$, such that $\forall$ strings $w \in L$, where $|w| \geq k$, $\exists$ u, v, x, y, z, such that: $w = uvxyz$, $vy \neq \varepsilon$, $|vxy| \leq k$, and $\forall q \geq 0$, uv^qxy^qz is in L.

Proof: L is generated by some CFG G = (V, Σ, R, S) with n nonterminal symbols and branching factor b. The longest string that can be generated by G with no repeated nonterminals in the resulting parse tree has length bn.

Let k be b^{n+1}. Assuming that $b \geq 2$, it must be the case that $b^{n+1} > b^n$. So, let w be any string in $L(G)$ where $|w| \geq k$.

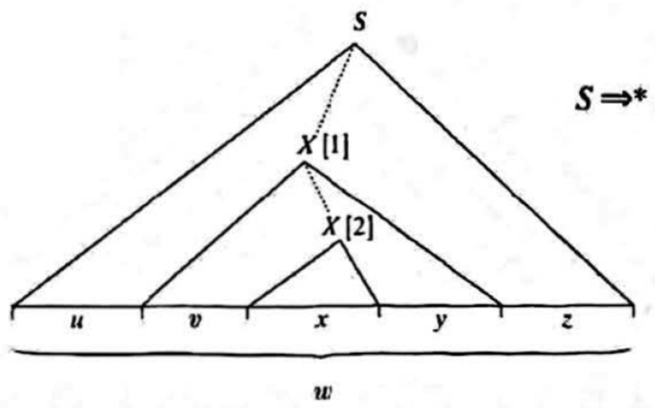
Let T be any smallest parse tree for w. T must have height at least $n + 1$. Choose some path in T of length at least $n + 1$.

Let X be the bottom-most repeated nonterminal along that path. Then w can be rewritten as $uvxyz$. The tree rooted at [1] has height at most $n + 1$. Thus its yield, vxy, has length less than or equal to b^{n+1}, which is k.

$vy \neq \varepsilon$ since if vy were ε then there would be a smaller parse tree for w and we chose T so that that wasn't so.

uxz must be in L because $rule_2$ could have been used immediately at [1]. For any $q \geq 1$, uv^qxy^qz must be in L because $rule_1$ could have been used q times before finally using $rule_2$.

So, if L *is* a context-free language, every "long" string in L must he pumpable. If there is even one long string in L that is not pumpable then L is not context-free.



$$S \Rightarrow^* uXz \Rightarrow^* uvXyz \Rightarrow^* uvxyz.$$

Regular vs CF Pumping Theorems

Similarities:

- We choose w, the string to be pumped.
- We choose a value for q that shows that w isn't pumpable.
- We may apply closure theorems before we start.

Differences:

- Two regions, v and y, must be pumped in tandem.
- We don't know anything about where in the strings v and y will fall. All we know is that they are reasonably "close together", i.e., $|vxy| \leq k$.
- Either v or y could be empty, although not both.

Prove that L = { $a^nb^nc^n$, n≥0} not context free

Let L = { $a^nb^nc^n$, $n \geq 0$}

We use the pumping theorem to show that L is not CFL. If it were, then there would exist some *k* such that any string *w,* where $|w| \geq k$, must satisfy the conditions of the theorem. We show one string *w* that does not.

Let $w = a^kb^kc^k$, where *k* is the constant from the Pumping Theorem.

For w to satisfy the conditions of the Pumping Theorem, there must be some u, v, x, y, and z such that w = uvxyz, $vy \neq \varepsilon$, $|vxy| \leq k$ and $\forall q \geq 0$, uv^qxy^qz is in L. We show that no such *u, v, x, y,* and *z* exist.

If either v or y contains two or more different characters, then set q to 2 (i.e. pump in once) and the resulting string will have letters out of order and thus not he in L (For example, if v is aabb and y is cc, then the string that results from pumping will look like aaa.. . aaabbaabbccc …ccc.)

If both v and y each contain almost one distinct character, then set q to 2. Additional copies of at most two different characters are added, leaving the third unchanged. There are no longer equal numbers of the three letters. So, the resulting string is not in L.

There is no way to divide *w* into *uvxyz* such that all the conditions of the Pumping Theorem are met. So, L is not context-free.

The Language of Strings with n^2 a's is not CFL

Proof: Let $L = \{a^{n^2} : n \geq 0\}$.

We use the pumping theorem to show that L is not CFL. If it were, then there would exist some *k* such that any string *w,* where $|w| \geq k$, must satisfy the conditions of the theorem. We show one string *w* that does not.

Let n (in the definition of L) be k^2. So $n^2 = k^4$ and $w = a^{k^4}$

For *w* to satisfy the conditions of the Pumping Theorem, there must be some *u, v, x, y,* and *z* such that w = *uvxyz*, $vy \neq \varepsilon$. $|vxy| \leq k$. and $\forall q \geq 0$, uv^qxy^qz is in L. We show that no such *u, v, x, y,* and *z* exist.

Since *w* contains only a's, *vy* = a^p, for some nonzero *p*. Set *q* to 2. The resulting string, $s = a^{k^4+p}$, which must be in *L*. But it isn't because it is too short.

- If a^{k^4}, which contains $(k^2)^2$ a's, is in *L*, then the next longer element of *L* contains $(k^2 + 1)^2 = k^4 + 2k^2 + 1$ a's.
- So, there are no strings in *L* with length between k^4 and $k^4 + 2k^2 + 1$.
- But $|s| = k^4 + p$. So, for *s* to be in *L* , $p = |vy|$ would have to be at least $2k^2 + 1$.
- But $|vxy| \leq k$, *so p* can't be that large.

Thus, s is not in *L*. There is no way to divide *w* into *uvxyz* such that all the conditions of the Pumping Theorem are met. So, *L* is not context-free.

Dividing the String w Into Regions

Prove that L = {$a^n b^m a^n$: m, n ≥ 0, m≥n} is not context free.

Let L = {$a^n b^m a^n$: m,n ≥ 0, m≥n}.

We use the pumping theorem to show that L is not CFL. If it were, then there would exist some *k* such that any string *w*, where |w| ≥ *k*, must satisfy the conditions of the theorem. We show one string *w* that does not.

Let w = $a^k b^k c^k$

For w to satisfy the conditions of the Pumping Theorem, there must be some u, v, x, y. and z, such that w = uvxyz, vy ≠ ε. lvxyl≤ k. and ∀ q ≥ 0 ($uv^q xy^q z$ is in L). We show that no such u, v, x, y, and z exist.

Imagine w divided into three regions as follows:

```
aaa ... aaabbb ... bbbaaa ...aaa
|    1    |    2    |    3    |
```

If either *v* or *y* crosses regions, then set *q* to 2 (thus pumping in once). The resulting string will have letters out of order and so not be in L. So in all the remaining cases we assume that *v* and y each falls within a single region.

- (1,1): Both *v* and y fall in region 1. Set *q to* 2. In the resulting string, the first group of a's is longer than the second group of a's. So, the string is not in *L*.
- (2. 2): Both *v* and y fall in region 2. Set *q* to 2. In the resulting string, the b region is longer than either of the a regions. So, the string is not in L
- (3, 3): Both *v* and y fall in region 3. Set *q* to 0. The same argument as for (1,1).
- (1, 2): Nonempty *v* falls in region 1 and nonempty *y* in region 2. Set *q* to 2. In the resulting string, the first group of a's is longer than the second group of a's. So, the string is not in *L*.
- (2, 3): Nonempty v falls in region 2 and nonempty *y* in region 3. Set *q* to 2. In the resulting string the second group of a's is longer than the first group of a's. So, the string is not in *L*.
- (1, 3): Nonempty *v* falls in region 1 and nonempty *y* in region 3. If this were allowed by the other conditions of the Pumping Theorem, we could pump in a's and still produce strings in *L*. But if we pumped out, we would violate the requirement that the a regions be at least as long as the b region. More importantly, this case violates the requirement that |vxy| ≤ *k* . So it need not be considered.

There is no way to divide w into *uvxyz* such that all the conditions of the Pumping Theorem are met. So, *L* is not context-free.

Prove L = {wcw, w is in {a,b}* } if not CFL

Proof: Let L = {wcw; w belongs to {a,b}* }.

We use the pumping theorem to show that L is not CFL. If it were, then there would exist some *k* such that any string *w,* where $|w| \geq k$, must satisfy the conditions of the theorem. We show one string *w* that does not.

Let $w = a^k b^k c a^k b^k$

For w to satisfy the conditions of the Pumping Theorem, there must be some u, v, x, y. and z, such that w = uvxyz, $vy \neq \varepsilon$. $|vxy| \leq k$. and $\forall q \geq 0$ ($uv^q xy^q z$ is in L). We show that no such u, v, x, y, and z exist.

Imagine w divided into three regions as follows:

```
aaa ... aaabbb ... bbbcaaa ... aaabbb ... bbb
|   1   |     2    |3|    4    |     5    |
```

Call the part before the c the left side and the part after the c the right side. We consider all the cases for where v and y could fall and show that in none of them are all the conditions of the theorem met:

- If either v or y overlaps region 3, set q to 0. The resulting string will no longer contain a c and so is not in L.
- If both v and y occur before region 3 or they both occur after region 3, then set q to 2. One side will be longer than the other and so the resulting string is not in L.
- If either v or y overlaps region 1, then set q to 2. 1n order to make the right-side match something would have to be pumped into region 4. But any v, y pair that did that would violate the requirement that $|vxy| \leq k$.
- If either v or y overlaps region 2, then set q to 2. In order to make the right-side match, something would have to be pumped into region 5. But any v, y pair that did that would violate the requirement that $|vxy| \leq k$.

There is no way to divide w into uvxyz such that all the conditions of the Pumping Theorem are met. So, L is not context-free.

Using the pumping theorem in conjunction with the closure Properties

Prove WW = {*ww, w* ∈ {a. b }*} is not context free

If WW were context-free, then $L' = WW \cap a^*b^*a^*b^*$ would also be context free. But it isn't, (a*b*a*b* is a regular language).

We use the pumping theorem to show that L' is not CFL. If it were, then there would exist some *k* such that any string *w,* where $|w| \geq k$, must satisfy the conditions of the theorem. We show one string *w* that does not.

Let $w = a^k\, b^k\, c\, a^k\, b^k$

For w to satisfy the conditions of the Pumping Theorem, there must be some u, v, x, y. and z, such that w = uvxyz, vy $\neq \varepsilon$. $|vxy| \leq k$. and $\forall q \geq 0$ (uv^qxy^qz is in L). We show that no such u, v, x, y, and z exist.

Imagine w divided into four regions as follows:

```
aaa ... aaabbb ... bbbaaa ... aaabbb ... bbb
|    1     |    2     |    3     |    4     |
```

We consider all the cases for where v and y could fall and show that in none of them are all the conditions of the theorem met:

- If either v or y overlaps more than one region, set q to 2. The resulting string will not be in a*b*a*b* and so is not in L'.
- If |vy| is not even then set q to 2. The resulting string will have odd length and so not be in L'. We assume in all the other cases that |vy| is even.
- (1, 1), (2, 2), (1, 2): Set q to 2. The boundary between the first half and the second half will shift into the first b region. So the second half will start with a **b**, while the first half still starts with an **a**. So the resulting string is not in L'.
- (3, 3), (4, 4), (3, 4): Set q to 2. This time the boundary shifts into the second a region. The first half will end with an a while the second half still ends with a b. So, the resulting string is not in L'.
- (2, 3): Set q to 2. If $|v| \neq |y|$ then the boundary moves and, as argued above the resulting string is not in L'. If |v| = |y| then the first half contains more b's and the second half contains more a's. Since they are no longer the same, the resulting string is not in L'.
- (1, 3), (1, 4), and (2, 4) violate the requirement that $|vxy| \leq k$.

There is no way to divide w into uvxyz such that all the conditions of the Pumping Theorem are met. So L' is not context-free. So neither is WW.

A Simple Arithmetic Language is Not Context-Free

Proof:

Let $L = \{x \# y = z : x, y, z \in \{0,1\}^*$ and, if x, y and z are viewed as positive binary numbers without leading zeros, then $xy = z^R\}$. For example, 100#111 = 00111 $\in L$. (We do this example instead of the more natural one in which we require that $xy = z$ because it seems as though it might be more likely to be context-free. As we'll see, however, even this simpler variant is not.)

If L were context-free, then $L' = L \cap 10^*\#1^* = 0^*1^*$ would also be context-free. But it isn't, which we can show using the Pumping Theorem. If it were, then there would exist some k such that any string w, where $|w| \geq k$, must satisfy the conditions of the theorem. We show one string w that does not. Let $w = 10^k\#1^k = 0^k1^k$, where k is the constant from the Pumping Theorem. Note that $w \in L$ because $10^k \cdot 1^k = 1^k0^k$.

For w to satisfy the conditions of the Pumping Theorem, there must be some u, v, x, y, and z, such that $w = uvxyz$, $vy \neq \varepsilon$, $|vxy| \leq M$, and $\forall q \geq 0$ (uv^qxy^qz is in L). We show that no such u, v, x, y, and z exist. Imagine w divided into seven regions as follows:

```
1 000 ... 000 # 111 ... 111  =  000 ... 000111 ... 111
|1|    2     |3|    4       |5|      6      |    7    |
```

We consider all the cases for where v and y could fall and show that in none of them are all the conditions of the theorem met:

- If either v or y overlaps region 1, 3, or 5 then set q to 0. The resulting string will not be in 10*#1* = 0*1* and so is not in L'.
- If either v or y contains the boundary between 6 and 7, set q to 2. The resulting string will not be in 10*#1* = 0*1* and so is not in L'. So the only cases left to consider are those where v and y each occur within a single region.
- (2, 2), (4, 4), (2, 4): Set q to 2. Because there are no leading zeros, changing the left side of the string changes its value. But the right side doesn't change to match. So the resulting string is not in L'.
- (6, 6), (7, 7), (6, 7): Set q to 2. The right side of the equality statement changes value but the left side doesn't. So the resulting string is not in L'.
- (4, 6): Note that, because of the first argument to the multiplication, the number of 1's in the second argument must equal the number of 1's after the =. Set q to 2. The number of 1's in the second argument changed but the number of 1's in the result did not. So the resulting string is not in L'.
- (2, 6), (2, 7), and (4, 7) violate the requirement that $|vxy| \leq k$.

There is no way to divide w into $uvxyz$ such that all the conditions of the Pumping Theorem are met. So L is not context-free.