بسم اللّه الرّحمن الرّحیم

دانشگاه صنعتی اصفهان ــ دانشکدهٔ مهندسی برق و کامپیوتر
(نیم‌سال تحصیلی ۴۰۲۱)

نظریهٔ زبان‌ها و ماشین‌ها

حسین فلسفین

Non-Context-Free Languages

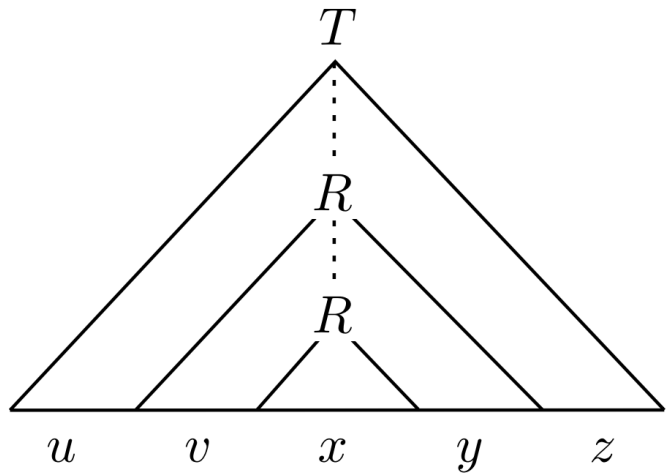
In this session, we present a technique for proving that certain languages are not context free. Recall that in the previous sessions we introduced the pumping lemma for showing that certain languages are not regular. Here we present a similar pumping lemma for context-free languages. It states that every context-free language has a special value called the pumping length such that all longer strings in the language can be "pumped." This time the meaning of pumped is a bit more complex. It means that the string can be divided into five parts so that the second and the fourth parts may be repeated together any number of times and the resulting string still remains in the language.

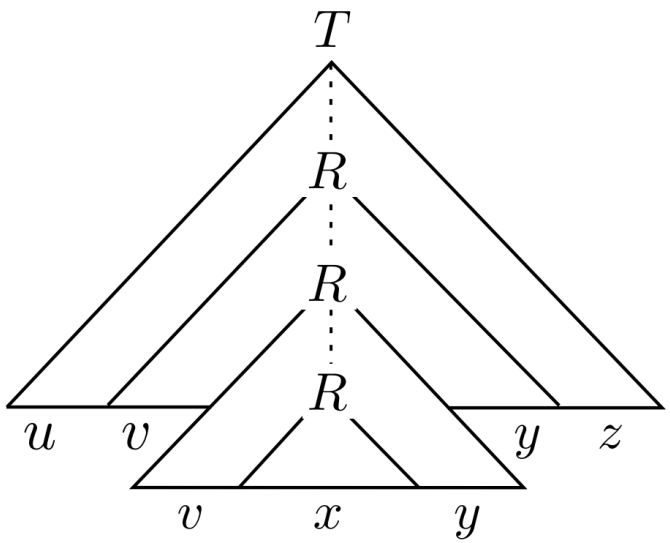
Theorem (Pumping lemma for context-free languages): *If A is a context-free language, then there is a number p (the pumping length) where, if s is any string in A of length at least p, then s may be divided into five pieces $s = uvxyz$ satisfying the conditions*

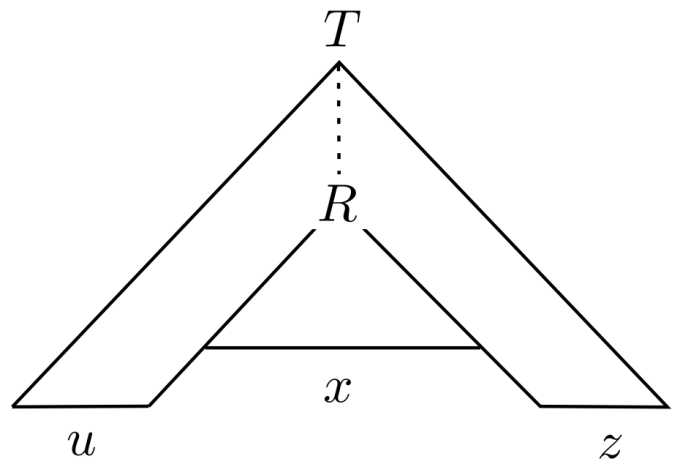
1. *for each $i \geq 0$, $uv^ixy^iz \in A$,*
2. *$|vy| > 0$, and*
3. *$|vxy| \leq p$.*

When s is being divided into $uvxyz$, condition 2 says that either v or y is not the empty string. Otherwise the theorem would be trivially true. Condition 3 states that the pieces v, x, and y together have length at most p.

Proof Idea: *Let A be a CFL and let G be a CFG that generates it. We must show that any sufficiently long string s in A can be pumped and remain in A. The idea behind this approach is simple. Let s be a very long string in A. (We make clear later what we mean by "very long.") Because s is in A, it is derivable from G and so has a parse tree. The parse tree for s must be very tall because s is very long. That is, the parse tree must contain some long path from the start variable at the root of the tree to one of the terminal symbols at a leaf. On this long path, some variable symbol R must repeat because of the pigeonhole principle. As the following figure shows, this repetition allows us to replace the subtree under the second occurrence of R with the subtree under the first occurrence of R and still get a legal parse tree. Therefore, we may cut s into five pieces $uvxyz$ as the figure indicates, and we may repeat the second and fourth pieces and obtain a string still in the language. In other words, uv^ixy^iz is in A for any $i \geq 0$.*


T
R
R
u
v
x
y
z


T
R
R
R
u
v
y
z
v
x
y


T
R
x
u
z

Proof: *Let G be a CFG for CFL A. Let b be the maximum number of symbols in the right-hand side of a rule (assume at least 2). In any parse tree using this grammar, we know that a node can have no more than b children. In other words, at most b leaves are 1 step from the start variable; at most b^2 leaves are within 2 steps of the start variable; and at most b^h leaves are within h steps of the start variable. So, if the height of the parse tree is at most h, the length of the string generated is at most b^h. Conversely, if a generated string is at least $b^h + 1$ long, each of its parse trees must be at least $h + 1$ high. Say $|V|$ is the number of variables in G. We set p, the pumping length, to be $b^{|V|+1}$. Now if s is a string in A and its length is p or more, its parse tree must be at least $|V| + 1$ high, because $b^{|V|+1} \geq b^{|V|} + 1$. To see how to pump any such string s, let τ be one of its parse trees. If s has several parse trees, choose τ to be a parse tree that has the smallest number of nodes. We know that τ must be at least $|V| + 1$ high, so its longest path from the root to a leaf*

has length at least $|V| + 1$. That path has at least $|V| + 2$ nodes; one at a terminal, the others at variables. Hence that path has at least $|V| + 1$ variables. With G having only $|V|$ variables, some variable R appears more than once on that path. For convenience later, we select R to be a variable that repeats among the lowest $|V| + 1$ variables on this path. We divide s into $uvxyz$ according to the above figure. Each occurrence of R has a subtree under it, generating a part of the string s. The upper occurrence of R has a larger subtree and generates vxy, whereas the lower occurrence generates just x with a smaller subtree. Both of these subtrees are generated by the same variable, so we may substitute one for the other and still obtain a valid parse tree.

☞ *Replacing the smaller by the larger repeatedly gives parse trees for the strings uv^ixy^iz at each $i > 1$. Replacing the larger by the smaller generates the string $uv^0xy^0z = uxz$. That establishes condition 1 of the lemma. We now turn to conditions 2 and 3.*

☞ *To get condition 2, we must be sure that v and y are not both ε. If they were, the parse tree obtained by substituting the smaller subtree for the larger would have fewer nodes than τ does and would still generate s. This result isn't possible because we had already chosen τ to be a parse tree for s with the smallest number of nodes. That is the reason for selecting τ in this way.*

☞ *In order to get condition 3, we need to be sure that vxy has length at most p. In the parse tree for s the upper occurrence of R generates vxy. We chose R so that both occurrences fall within the bottom $|V|+1$ variables on the path, and we chose the longest path in the parse tree, so the subtree where R generates vxy is at most $|V|+1$ high. A tree of this height can generate a string of length at most $b^{|V|+1} = p$.*

Example 1: *Use the pumping lemma to show that the language $B = \{a^n b^n c^n | n \geq 0\}$ is not context free. We assume that B is a CFL and obtain a contradiction. Suppose, to the contrary, that B is a CFL. Let p be the pumping length for B that is guaranteed to exist by the pumping lemma. Select the string $s = a^p b^p c^p$. Clearly s is a member of B and of length at least p. The pumping lemma states that s can be pumped, but we show that it cannot. We show that no matter how we divide s into $uvxyz$, one of the three conditions of the lemma is violated. By condition 3, vxy is either a substring of $a^p b^p$ or a substring of $b^p c^p$. If vxy is a substring of $a^p b^p$ then, by condition 1, the string uxz has either less than p occurrences of a or less than p occurrences of b. However, since vxy is a substring of $a^p b^p$, we know that c^p is a substring of z and so uxz has p occurrences of symbol c. It follows that $uxz \notin B$, contradicting condition 1. Similarly, we can also get a contradiction in the case that vxy is a substring of $b^p c^p$.*

Example 2: *Show that* $L = \{a^i b^j c^k : k = \max\{i, j\}\}$ *is not a context-free language.*

Proof. *Suppose, to the contrary, that* L *is a context-free language. Let* p *be the pumping length given by the pumping lemma. Consider string* $s = a^p b^p c^p$*. The string* s *can be decomposed as* $s = uvxyz$*, satisfying conditions 1–3 of the lemma. By condition 3, we know that* vxy *cannot contain both* a *and* c*.*

Case 1. vy *contains symbol* c*. In this case, the number of* c*'s in* uxz *is less than* p *and the number of* a*'s in* uxz *is equal to* p*. Hence,* $uxz \notin L$*, which contradicts condition 1.*

Case 2. vy *does not contain symbol* c*. In this case, either the number of* a*'s or the number of* b*'s in* uv^2xy^2z *is greater than* p *and the number of* c*'s in* uv^2xy^2z *is equal to* p*. Hence,* $uv^2xy^2z \notin L$*, again a contradiction.*

Example 3: *Let $C = \{a^i b^j c^k | 0 \leq i \leq j \leq k\}$. We use the pumping lemma to show that C is not a CFL. Suppose, to the contrary, that C is a context-free language. Let p be the pumping length given by the pumping lemma. Consider string $s = a^p b^p c^p$. The string s can be decomposed as $s = uvxyz$, satisfying conditions 1–3 of the lemma. By condition 3, we know that vxy cannot contain both a and c.*

Case 1. *vy contains symbol c. In this case, the number of c's in uxz is less than p and the number of a's in uxz is equal to p. Hence, $uxz \notin C$, which contradicts condition 1.*

Case 2. *vy does not contain symbol c. In this case, either the number of a's or the number of b's in uv^2xy^2z is greater than p and the number of c's in uv^2xy^2z is equal to p. Hence, $uv^2xy^2z \notin C$, again a contradiction.*

Example 4: Applying the Pumping Lemma to

$$\{x \in \{a,b,c\}^* | n_a(x) < n_b(x) \text{ and } n_a(x) < n_c(x)\}$$

Let $L = \{x \in \{a,b,c\}^* | n_a(x) < n_b(x)$ ***and*** $n_a(x) < n_c(x)\}$***, suppose that*** L ***is a CFL, and let*** p ***be the integer in the pumping lemma. Let*** $s = a^p b^{p+1} c^{p+1}$***, and let*** u***,*** v***,*** x***,*** y***, and*** z ***be strings for which*** $s = uvxyz$ ***and conditions 1–3 are satisfied.***

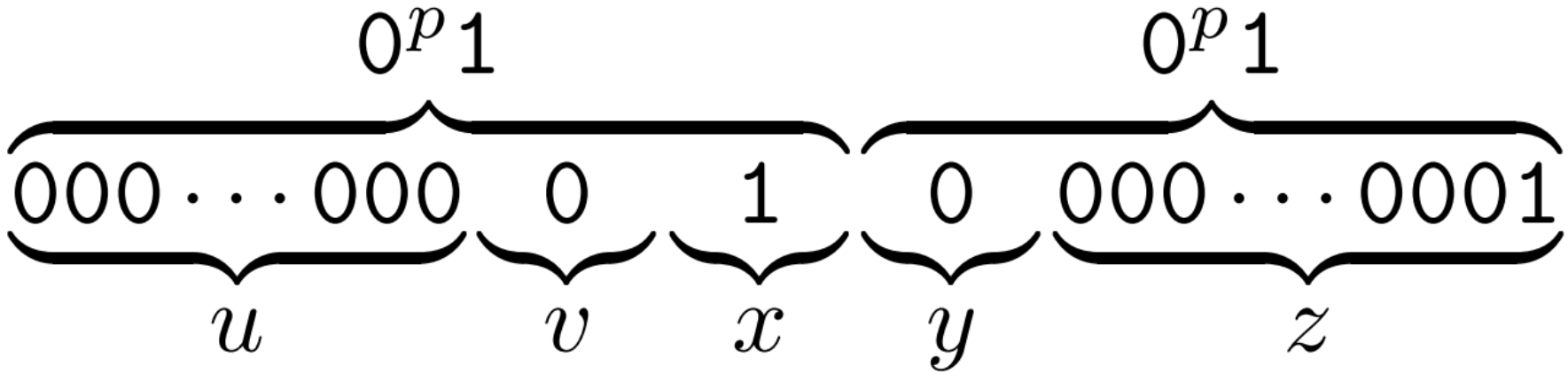
☞ ***If*** vy ***contains an*** a***, then because of condition 3 it cannot contain a*** c***. It follows that*** uv^2xy^2z***, obtained by adding one copy of*** v ***and*** y ***to*** s***, has at least*** $p+1$ a***'s and exactly*** $p+1$ c***'s.***

☞ ***On the other hand, if*** vy ***does not contain an*** a***, then it contains either a*** b ***or a*** c***; in this case,*** uv^0xy^0z ***contains exactly*** p a***'s, and either no more than*** p b***'s or no more than*** p c***'s. In both cases, we obtain a contradiction.***

Example 5: *The language $L = \{a^i b^j c^i d^j | i, j \geq 1\}$ is not a context-free language.*

Proof: *Suppose that L is a CFL, and let p be the integer in the pumping lemma. Consider $s = a^p b^p c^p d^p \in L$. The string s can be decomposed as $s = uvxyz$, satisfying conditions 1–3 of the lemma. By condition 3, we know that $|vxy| \leq p$. Since $|vxy| \leq p$, it must be contained in a power of one of the letters a, b, c, or d or in the powers of two adjacent letters. If it is contained in the power of one letter, say a, then the number of a's in uxz is less than p and the number of c's in uxz is equal to p. Hence, $uxz \notin L$, which contradicts condition 1. If it is contained in a power of two adjacent letters, say a and b, then either the number of a's in uxz is less than p and the number of c's in uxz is equal to p, or the number of b's in uxz is less than p and the number of d's in uxz is equal to p (or both). Hence, $uxz \notin L$, again a contradiction.*

Example 6: ***Let $D = \{ww | w \in \{0,1\}^*\}$. Use the pumping lemma to show that D is not a CFL. Assume that D is a CFL and obtain a contradiction. Let p be the pumping length given by the pumping lemma. This time choosing string s is less obvious. One possibility is the string 0^p10^p1. It is a member of D and has length greater than p, so it appears to be a good candidate. But this string can be pumped by dividing it as follows, so it is not adequate for our purposes.***



Let's try another candidate for s. Intuitively, the string $0^p1^p0^p1^p$ seems to capture more of the "essence" of the language D than the previous candidate did. In fact, we can show that this string does work, as follows. We show that the string $s = 0^p1^p0^p1^p$ cannot be pumped. By condition 3, vxy is a substring of the first half of ww (0^p1^p), or a substring of the middle $2p$ letters of ww (1^p0^p), or a substring of the second half of ww (0^p1^p). In each case, uxz contains a block of 0's (or 1's) that is shorter than p, and a block of 0's (or 1's, respectively) of length exactly p. It is obvious that such a string is not equal to $w'w'$ for any $w' \in \{0,1\}^*$. This contradicts condition 1 and, hence, proves that D is not context-free.

Example 7: ***The language*** $L = \{x \in \{1\}^* : \textbf{the length of } x \textbf{ is a prime}\}$ ***is not context-free.***

فرض کنید (فرض خلف) که این زبان **CFL** باشد. اگر p همان عددی باشد که لم تزریق وعدهٔ وجود آن‌را می‌دهد، آنگاه ما رشتهٔ $s = 1^\alpha \in L$ را در نظر می‌گیریم، که در آن، α یک عدد اول بزرگتر از $p+1$ است. حال فرض کنید که $s = uvxyz$ همان تجزیه‌ای باشد که لم، وجودش را تضمین می‌کند. اکنون قرار دهید $\beta = |uv^0xy^0z| = |uxz|$. حال داریم

$$|uv^\beta xy^\beta z| = |uxz| + \beta|vy| = |uxz| + \beta(|s| - |uxz|) = \beta + \beta(\alpha - \beta).$$

اما این عدد، یعنی $\beta(\alpha - \beta + 1)$ مرکب است. چرا؟ چون اولاً،

$$\beta = |uxz| = |s| - |vy| \geq p + 2 - |vy| \geq p + 2 - |vxy| \geq 2,$$

و ثانیاً، $\alpha - \beta + 1 = |vy| + 1 \geq 2$. لذا، حاصل پامپ کردن رشتهٔ s، در زبان L قرار نمی‌گیرد. این با شرط اول لم در تعارض است. پس فرض خلف باطل است.

Example 8: ***The language*** $L = \{a^{n^2}, n \geq 0\}$ ***is not context-free. If it were, then there would exist some*** p ***such that any string*** s***, where*** $|s| \geq p$***, must satisfy the conditions of the lemma. We show one string*** s ***that does not. Let*** n ***(in the definition of*** L***) be*** p^2***. So*** $n^2 = p^4$ ***and*** $s = a^{p^4}$***. For*** s ***to satisfy the conditions of the pumping lemma, there must be some*** u***,*** v***,*** x***,*** y***, and*** z***, such that*** $s = uvxyz$***,*** $vy \neq \varepsilon$***,*** $|vxy| \geq p$***, and*** $\forall i \geq 0$ ***(*** uv^ixy^iz ***is in*** L***). We show that no such*** u***,*** v***,*** x***,*** y***, and*** z ***exist. Since*** s ***contains only*** a***'s,*** $vy = a^k$***, for some nonzero*** k***. Set*** i ***to*** 2***. The resulting string, which we'll call*** s'***, is*** a^{p^4+k}***, which must be in*** L***. But it isn't because it is too short. If*** a^{p^4}***, which contains*** $(p^2)^2$ a***'s, is in*** L***, then the next longer element of*** L ***contains*** $(p^2+1)^2$ a***'s. That's*** p^4+2p^2+1 a***'s. So there are no strings in*** L ***with length between*** p^4 ***and*** $p^4 + 2p^2 + 1$***. But*** $|s'| = p^4 + k$***. So, for*** s' ***to be in*** L***,*** $k = |vy|$ ***would have to be at least*** $2p^2 + 1$***. But*** $|vxy| \leq p < 2p^2 + 1$***, so*** k ***can't be that large. Thus*** s' ***is not in*** L***.***

ذکر یک نکتهٔ مهم

اگر یک زبان، مستقل از متن باشد، آنگاه لم تزریق قطعاً برای آن برقرار است.

اما اگر لم تزریق برای یک زبان برقرار باشد، آنگاه آن زبان الزاماً مستقل از متن نیست. لذا برای اثبات مستقل از متن بودن یک زبان، نمی‌توان از برقرار بودن لم تزریق برای آن بهره گرفت.

Problem 2.36: ***Give an example of a language that is not context free but that acts like a CFL in the pumping lemma. Prove that your example works. (See the analogous example for regular languages in Problem 1.54.)***

تمرین: زبان

$$L = \{a^k b^l c^m : m = kl\}$$

(روی الفبای $\{a, b, c\}$) را در نظر بگیرید. قرار است با بهره‌گیری از لم تزریق نشان دهیم که این زبان، یک زبان مستقل از متن نیست.

الف. چرا رشتهٔ $s = ab^p c^p$ برای رسیدن به تناقض مناسب نیست؟

ب. با بهره‌گیری از رشتهٔ $s' = a^p b^p c^{p^2}$ نشان دهید که این زبان مستقل از متن نیست. **راهنمایی:** دو حالت را مدنظر قرار دهید: اینکه vy دربردارندهٔ سمبل a باشد یا نباشد. در هر دو صورت به تناقض برسید.

Using the Pumping Lemma in Conjunction with the Closure Properties

Although context-free languages are not closed under intersection, we know that the intersection of a context-free language and a regular language must be context-free. This weaker closure property is still helpful in proving a language not context-free.

Example: *Show that* $L = \{w \in \{a, b, c\}^* | n_a(w) = n_b(w) = n_c(w)\}$ *is not context-free.*

Proof. *We note that the language* $L_1 = \{a^n b^n c^n | n \geq 0\}$ *is equal to* $L \cap L(a^* b^* c^*)$*. We know that* L *cannot be context-free, for otherwise* L_1 *would also be context-free, contradicting our previous example.*

Ogden's Lemma

The pumping lemma is one of the most important tools we have for proving languages not context-free. Here, we state, without proof, a more powerful version of the pumping lemma known as Ogden's lemma.

Suppose A is a context-free language. Then there is an integer p so that for every $s \in A$ with $|s| \geq p$, and every choice of p or more "distinguished (marked)" positions in the string s, there are strings u, v, x, y, and z so that $s = uvxyz$ and the following conditions are satisfied.

1. *For every $i \geq 0$, $uv^ixy^iz \in A$.*
2. *The string vy contains at least one distinguished position.*
3. *The string vxy contains p or fewer distinguished positions.*