The Pumping Lemma for Context-Free Languages

Lecture 13

November 27, 2022

Objectives

By the end of this lecture, you should be able to

- Use the pumping lemma to prove that a language is not context-free.

Non-Context-Free Languages

Some simple languages are not context-free.

1. $L_1 = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n | \ n \in \mathcal{N}\}$.
2. $L_2 = \{ww | \ w \in \{\mathtt{a}, \mathtt{b}\}\}$.
3. $L_3 = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k | \ 0 \leq i \leq j \leq k\}$

Whatever do we mean by simple?

Proving Languages Non-Context-Free

- Recall:
  - To prove that a language is regular, construct an NFA that recognizes it.
  - To prove that a language is not regular, use the pumping lemma for regular languages.
- To prove that a language is CF, construct a CFG or a PDA.
- To prove that a language is not CF, use the pumping lemma for CFLs.
  - This does not always work (same with regular languages). Some non-CFLs observe the pumping lemma for CFLs.
  - After studying this lecture, can you give an example of one?
  - What would you do to prove that a language is neither regular nor context-free?

A Note on Trees

- Consider a tree where each node has a maximum of b children.
  - At level zero, there is one node: the root.
  - At level one, there are (at most) b nodes.
  - At level n, there are (at most) b^n nodes.
- Thus, the maximum number of nodes at level l is b^l.
- Hence, if the number of leaves of a tree is at least $b^l + 1$, then its height is at least $l + 1$.
- It follows that, for such a tree, the longest path from the root to a leaf has at least $l + 2$ nodes on it (counting the root).
- Hence the number of internal nodes on such a path is at least $l + 1$.

A Note on *Parse* Trees

- Let G be a CFG and s a string in $L(G)$.
- s has a parse tree depicting a derivation in G.
- The maximum number of children of a node in the parse tree is equal to the length b of the longest right-hand side of a rule of G.
- If $|s| \geq b^l + 1$, then the height of the parse tree is at least $l + 1$.
- It follows that the longest path from the root (S) to a leaf has at least $l + 2$ nodes on it.
- Hence the number of variables (i.e., internal nodes) on such a path is at least $l + 1$.

The Pumping Lemma for CFLs

Theorem (Sipser 2.34) If A is a context-free language, then there is a number p (the pumping length) where, if s is any string in A with length at least p, then s may be divided into five pieces, $s = uvxyz$, satisfying the following conditions:

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

Basic Insight

The proof, again, is based on the pigeonhole principle.

- Suppose that A is a CFL.
- Thus, there is a grammar G, such that $L(G) = A$.
- Let b be the length of a longest right-hand side of a rule of G.
  - What happens when $b \leq 1$?
- For any $s \in A$, if $|s| \geq b^l + 1$, then the number of variables (internal nodes of a parse tree) in the longest path from the root to a leaf is at least $l + 1$.
- If $l = |V|$, it follows that at least one variable is repeated along such a path.
- Hence, we set p (the pumping length) to $b^{|V|+1} (> b^{|V|} + 1)$.

Sketch of a Proof

- Take s to be a string in A such that $|s| \geq b^{|V|+1}$.
- Let τ be a parse tree for s that has the smallest number of nodes.
- Let R be a variable that repeats on the longest path from the root to a leaf of τ.
- For convenience, take R to be a variable that repeats among the lowest $|V| + 1$ variables on this path. (You'll see why.)
- Divide s into $uvxyz$.
- The upper occurrence of R has a larger subtree that generates vxy.
- The lower occurrence of R has a smaller subtree that generates x.

Proving the First Condition

- Replacing one subtree by another should still give us a parse tree for a string in A.
- Replacing the smaller by the larger repeatedly gives parse trees for the strings uv^ixy^iz for each $i > 1$.
- Replacing the larger by the smaller gives a parse tree for uxz.
- This establishes the first condition of the pumping lemma.

Proving the Second Condition

- The second condition states that $|vy| > 0$.
- To establish the second condition, we need to show that v and y cannot both be ε.
- Suppose that $v = y = \varepsilon$.
- Thus, replacing the larger subtree rooted at R by the smaller subtree would yield a parse tree, for s, with fewer nodes.
- This contradicts our assumption that τ is the smallest parse tree for s.
- Hence v and y cannot both be ε.

Proving the Third Condition

- The third condition states that $|vxy| \leq p$.
- Suppose that $|vxy| > p$. Recall that $p = b^{|V|+1}$.
- Thus, the larger subtree rooted at R and generating vxy is at least $|V| + 2$ high.
- But we've assumed that both occurrences of R fall among the bottom $|V| + 1$ variables on the longest path in τ.
- Thus, the upper occurrence of R can be the root of a subtree which is at most $|V| + 1$ high.
- This leads to a contradiction.
- Hence $|vxy| \leq p$.

Example 1

Prove that $L = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n | \ n \in \mathcal{N}\}$ is not context-free.

Example 1

Prove that $L = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n | \ n \in \mathcal{N}\}$ is not context-free.

- Assume that L is context-free, and let p be the pumping length.

Example 1

Prove that $L = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n | \ n \in \mathcal{N}\}$ is not context-free.

- Assume that L is context-free, and let p be the pumping length.
- Let $s = \mathtt{a}^p\mathtt{b}^p\mathtt{c}^p$.

Example 1

Prove that $L = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n | \ n \in \mathcal{N}\}$ is not context-free.

- Assume that L is context-free, and let p be the pumping length.
- Let $s = \mathtt{a}^p\mathtt{b}^p\mathtt{c}^p$.
- Note:
  - $s \in L$.
  - $|s| = 3p > p$.

Example 1

Prove that $L = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n | \ n \in \mathcal{N}\}$ is not context-free.

- Assume that L is context-free, and let p be the pumping length.
- Let $s = \mathtt{a}^p\mathtt{b}^p\mathtt{c}^p$.
- Note:
  - $s \in L$.
  - $|s| = 3p > p$.
- By the pumping lemma, $\mathtt{a}^p\mathtt{b}^p\mathtt{c}^p = uvxyz$ such that
  1. for each $i \geq 0$, $uv^ixy^iz \in L$,
  2. $|vy| > 0$, and
  3. $|vxy| \leq p$.

Example 1 (Cont'd)

- By condition 3, vxy can contain at most two types of symbols (never as, bs, and cs).

Example 1 (Cont'd)

- By condition 3, vxy can contain at most two types of symbols (never `a`s, `b`s, and `c`s).
- Hence, pumping up or down will change the number of occurrences of only two of symbols.

Example 1 (Cont'd)

- By condition 3, vxy can contain at most two types of symbols (never `a`s, `b`s, and `c`s).
- Hence, pumping up or down will change the number of occurrences of *only two* of symbols.
- Thus, the resulting string will have p occurrences of one symbol and a different number of occurrences of at least one other symbol.

Example 1 (Cont'd)

- By condition 3, vxy can contain at most two types of symbols (never `a`s, `b`s, and `c`s).
- Hence, pumping up or down will change the number of occurrences of *only two* of symbols.
- Thus, the resulting string will have p occurrences of one symbol and a different number of occurrences of at least one other symbol.
- This string cannot be in the form $\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n$ and is, hence, not in L.

Example 1 (Cont'd)

- By condition 3, vxy can contain at most two types of symbols (never `a`s, `b`s, and `c`s).
- Hence, pumping up or down will change the number of occurrences of *only two* of symbols.
- Thus, the resulting string will have p occurrences of one symbol and a different number of occurrences of at least one other symbol.
- This string cannot be in the form $\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n$ and is, hence, not in L.
- It follows that L is not context-free.

Example 2

Prove that $L = \{ww |\ w \in \{0,1\}^*\}$ is not context-free.

Example 2

Prove that $L = \{ww |\ w \in \{0,1\}^*\}$ is not context-free.

- Assume that L is context-free, and let p be the pumping length.

Example 2

Prove that $L = \{ww |\ w \in \{0,1\}^*\}$ is not context-free.

- Assume that L is context-free, and let p be the pumping length.
- Let $s = 0^p 1^p 0^p 1^p$.

Example 2

Prove that $L = \{ww |\ w \in \{0,1\}^*\}$ is not context-free.

- Assume that L is context-free, and let p be the pumping length.
- Let $s = 0^p 1^p 0^p 1^p$.
- Note:
  - $s \in L$.
  - $|s| = 4p > p$.

Example 2

Prove that $L = \{ww |\ w \in \{0,1\}^*\}$ is not context-free.

- Assume that L is context-free, and let p be the pumping length.
- Let $s = 0^p 1^p 0^p 1^p$.
- Note:
  - $s \in L$.
  - $|s| = 4p > p$.
- By the pumping lemma, $0^p 1^p 0^p 1^p = uvxyz$ such that
  1. for each $i \geq 0$, $uv^i xy^i z \in L$,
  2. $|vy| > 0$, and
  3. $|vxy| \leq p$.

Example 2 (Cont'd)

- By condition 3, vxy must be in one of the following forms: $\mathtt{0}^j$, $\mathtt{1}^j$, $\mathtt{0}^j\mathtt{1}^k$, or $\mathtt{1}^j\mathtt{0}^k$.

Example 2 (Cont'd)

- By condition 3, vxy must be in one of the following forms: $\mathtt{0}^j$, $\mathtt{1}^j$, $\mathtt{0}^j\mathtt{1}^k$, or $\mathtt{1}^j\mathtt{0}^k$.
  1. $vxy = \mathtt{0}^j$: Pumping up (or down) results in a string of the form $\mathtt{0}^k\mathtt{1}^p\mathtt{0}^l\mathtt{1}^p$, where $k \neq l$.

Example 2 (Cont'd)

- By condition 3, vxy must be in one of the following forms: $\mathtt{0}^j$, $\mathtt{1}^j$, $\mathtt{0}^j\mathtt{1}^k$, or $\mathtt{1}^j\mathtt{0}^k$.
  1. $vxy = \mathtt{0}^j$: Pumping up (or down) results in a string of the form $\mathtt{0}^k\mathtt{1}^p\mathtt{0}^l\mathtt{1}^p$, where $k \neq l$.
  2. $vxy = \mathtt{1}^j$: Pumping up (or down) results in a string of the form $\mathtt{0}^p\mathtt{1}^k\mathtt{0}^p\mathtt{1}^l$, where $k \neq l$.

Example 2 (Cont'd)

- By condition 3, vxy must be in one of the following forms: $\mathtt{0}^j$, $\mathtt{1}^j$, $\mathtt{0}^j\mathtt{1}^k$, or $\mathtt{1}^j\mathtt{0}^k$.
  1. $vxy = \mathtt{0}^j$: Pumping up (or down) results in a string of the form $\mathtt{0}^k\mathtt{1}^p\mathtt{0}^l\mathtt{1}^p$, where $k \neq l$.
  2. $vxy = \mathtt{1}^j$: Pumping up (or down) results in a string of the form $\mathtt{0}^p\mathtt{1}^k\mathtt{0}^p\mathtt{1}^l$, where $k \neq l$.
  3. $vxy = \mathtt{0}^j\mathtt{1}^k$: Pumping down results in a string whose second half starts with a $\mathtt{1}$, or whose first half ends with a $\mathtt{0}$. (Do you see why?)

Example 2 (Cont'd)

- By condition 3, vxy must be in one of the following forms: $\mathtt{0}^j$, $\mathtt{1}^j$, $\mathtt{0}^j\mathtt{1}^k$, or $\mathtt{1}^j\mathtt{0}^k$.
  1. $vxy = \mathtt{0}^j$: Pumping up (or down) results in a string of the form $\mathtt{0}^k\mathtt{1}^p\mathtt{0}^l\mathtt{1}^p$, where $k \neq l$.
  2. $vxy = \mathtt{1}^j$: Pumping up (or down) results in a string of the form $\mathtt{0}^p\mathtt{1}^k\mathtt{0}^p\mathtt{1}^l$, where $k \neq l$.
  3. $vxy = \mathtt{0}^j\mathtt{1}^k$: Pumping down results in a string whose second half starts with a $\mathtt{1}$, or whose first half ends with a $\mathtt{0}$. (Do you see why?)
  4. $vxy = \mathtt{1}^j\mathtt{0}^k$: Pumping down results in a string of the form $\mathtt{0}^p\mathtt{1}^l\mathtt{0}^q\mathtt{1}^p$, where at least one of l and q is less than p. (What happens if you pump up?)

Example 2 (Cont'd)

- By condition 3, vxy must be in one of the following forms: $\mathtt{0}^j$, $\mathtt{1}^j$, $\mathtt{0}^j\mathtt{1}^k$, or $\mathtt{1}^j\mathtt{0}^k$.
  1. $vxy = \mathtt{0}^j$: Pumping up (or down) results in a string of the form $\mathtt{0}^k\mathtt{1}^p\mathtt{0}^l\mathtt{1}^p$, where $k \neq l$.
  2. $vxy = \mathtt{1}^j$: Pumping up (or down) results in a string of the form $\mathtt{0}^p\mathtt{1}^k\mathtt{0}^p\mathtt{1}^l$, where $k \neq l$.
  3. $vxy = \mathtt{0}^j\mathtt{1}^k$: Pumping down results in a string whose second half starts with a $\mathtt{1}$, or whose first half ends with a $\mathtt{0}$. (Do you see why?)
  4. $vxy = \mathtt{1}^j\mathtt{0}^k$: Pumping down results in a string of the form $\mathtt{0}^p\mathtt{1}^l\mathtt{0}^q\mathtt{1}^p$, where at least one of l and q is less than p. (What happens if you pump up?)
- In all cases, the resulting string is not in L. Hence, L cannot be context-free.

Poetically [a]

Any regular language L has a magic number p
And any long-enough word in L has the following property:
Amongst its first p symbols is a segment you can find
Whose repetition or omission leaves x amongst its kind.

So if you find a language L which fails this acid test,
And some long word you pump becomes distinct from all the rest,
By contradiction you have shown that language L is not
A regular guy, resiliant to the damage you have wrought.

But if, upon the other hand, x stays within its L,
Then either L is regular, or else you chose not well.
For w is xyz, and y cannot be null,
And y must come before p symbols have been read in full.

As mathematical postscript, an addendum to the wise:
The basic proof we outlined here does certainly generalize.
So there is a pumping lemma for all languages context-free,
Although we do not have the same for those that are r.e.

[a] By Harry Mairson http://www.cs.brandeis.edu/~mairson/poems/node1.html

Next time

- Turing Machines.

Points to take home

- Statement of the pumping lemma for CFLs.
- Proof of the pumping lemma for CFLs.
- Using the pumping lemma for CFLs.