Pumping Lemma - 1

- The first 9 slides were written out on paper during the class.
- Consider \( L_1 = \{a^ibc^j : i+j \text{ is prime, } |i - j| < 5\} \)
- The PL4CFL gives an "n" for this language if it were a CFL
- Take a prime \( p > n \) and then \( i = \lfloor p/2 \rfloor \) and \( j = \lceil p/2 \rceil \) \((i + j = p)\)
- Then \( a^ibc^j = uvwx^y \) and we can pump on \( v, x \)

Pumping Lemma - 2

- If \( v \) or \( x \) contained \( b \), then \( uv^2wx^2y \) would have 2 \( b \)'s and would not be in the language
- Otherwise, consider \( uv^{p+1}wx^{p+1}y \).
- Here, the number of \( a \)'s + the number of \( c \)'s is \( p + p |vx| = p(1 + |vx|) \), which is not prime

Pumping Lemma - 3

- Next, consider \( L_2 = \{a^{2i+1}z : i \geq 0, z \in (b + c)^*, \ n_b(z) = 4j + 3, \ n_c(z) = 3k, \ 2i - 3 > j, \ 2j < 3k \} \)
- We pick a string of the form \( a^ib^jc^t \in L_2 \) even though \( L_2 \) does not require the \( b \)'s precede all the \( c \)'s
- First, consider \( 2j < 3k \) : we could use \( j = 3n, k = 2n+1, \) then \( 2j = 6n < 3k = 6n+3 \)
- Next, look at \( 2i - 3 > j \)
Pumping Lemma - 4

- So far, we need $2i - 3 > j$, $2j < 3k$
- Because of “$2i$”, we move to $j = 6n$, then $k = 4n + 1$, $i = 3n + 2$ is the smallest that gives $2i - 3 > 6n$
- We always work near the edge of the region given by the inequality

![Diagram]

- We have picked the string $a^{6n+5}b^{24n+3}c^{12n+3} = uvwxy$ using PL4CFL

Pumping Lemma - 5

- If $v$ and $x$ are in the $a$’s, pump down to $uvw$. We get $a^{6n+5-|vx|}b^{24n+3}c^{12n+3}$
- If $|vx|$ is odd, then # $a$’s becomes even
- If $|vx|$ is even, then it is $> 2$ and the value of $i$ in $a^{2i+1}$ must have decreased by at least 1
- We look at the effect this change has on the constraint $2i - 3 > j$

Pumping Lemma - 6

- When $i = 3n + 2$, we had $2i - 3 = 6n + 1$ but now $i$ is reduced, so $i < 3n + 1$: $2i - 3 < 6n - 1 < 6n = j$ (contradiction)
- If $v$ and $x$ are in the $b$’s, just pump up as much as needed
- If $v$ and $x$ are in the $c$’s, pump down:
  - If $|vx|$ not a multiple of 3, $uvw \not\in L_2$
  - If $|vx| = 3r$, $n_c(uvw) = 3(k - r)$.
- Before, $k$ was $4n+1$ and $2j < 3k$, but now $k_{new} \leq 4n$ and $2j = 12n > 3k_{new}$

Pumping Lemma - 7

- If $v$ and $x$ contains $a$’s and $b$’s, pump up to break the constraint $2j < 3k$
- If $v$ and $x$ contains $b$’s and $c$’s, pump up to break the constraint $2i - 3 > j$
Pumping Lemma - 8

- If we have
  \[ a \ldots ab \ldots bc \ldots c \]
  \[ u \ v \ w \ x \ y \]
- Then \( uv^2wx^2y \) has the form
  \[ a \ldots ab \ldots bc \ldots c \]
  \[ u \ v \ w \ x \ y \]
- This is not in \( L_2 \) because of the format (a’s after some b’s)

Pumping Lemma - 9

- If we have
  \[ a \ldots ab \ldots bc \ldots c \]
  \[ u \ v \ w \ x \ y \]
- Then \( uv^3wx^3y \) has the form
  \[ a \ldots ab \ldots bc \ldots bc \ldots bc \ldots c \]
  \[ u \ v \ w \ x \ x \ x \ y \]
- This could be in \( L_2 \) because the order of the b’s and c’s does not matter

Computability

- Post’s Correspondence Problem
- Valid Turing Computations and other results

Instance of PCP

- Take two lists of strings over the same alphabet \( \Sigma \): \( A = w_1, w_2, \ldots, w_k \) and \( B = x_1, x_2, \ldots, x_k \) for the same \( k \)
- The two lists \( (A, B) \) are called an instance of Post’s Correspondence Problem (PCP)
The PCP instance has a solution when we can find a SINGLE sequence of indices
\( i_1, i_2, ..., i_m \) \( m \geq 1 \), such that
\[ i_1 w_{i_2} w_{i_3} \cdots w_{i_m} = x_{i_1} x_{i_2} x_{i_3} \cdots x_{i_m} \]

- Note that the sequence chosen is the same on both sides.

Example

- The PCP \( A = 1, 10111, 10 \) and \( B = 111, 10, 0 \) has a solution \( 2,1,1,3: 10111 1 1 10 = 10 111 111 0 \)

- The PCP \( A = 10, 011, 101 \) and \( B = 101, 11, 011 \) does not have a solution

PCP is undecidable

- A lengthy proof shows that if PCP were decidable then \( L_u \) would be recursive (a special form of PCP is used)

- PCP can be used to show that it is undecidable whether a CFG is ambiguous:

The CFG ambiguity problem

- Take any PCP over alphabet \( \Sigma: A = w_1, w_2, ..., w_n \) and \( B = x_1, x_2, ..., x_n \) for some \( n \)

- Let \( a_1, a_2, ..., a_n \) be \( n \) new symbols

- Define two new languages
**L_A and L_B**

- \( L_A = \{ w_{i_1}w_{i_2} \ldots w_{i_m}a_{i_m}a_{i_{m-1}} \ldots a_{i_1} \mid m \geq 1 \} \) and
- \( L_B = \{ x_{i_1}x_{i_2} \ldots x_{i_m}a_{i_m}a_{i_{m-1}} \ldots a_{i_1} \mid m \geq 1 \} \)
- These are infinite languages because we consider every \( m \) and every sequence \( i_1, i_2, \ldots, i_m \)

**CFG's**

- \( (\{S_A\}, \Sigma', P_A, S_A) \) generates \( L_A \) where \( P_A \) has productions \( S_A \rightarrow w_iS_Aa_i \) and \( S_A \rightarrow w_ia_i \)
- \( (\{S_B\}, \Sigma', P_B, S_B) \) generates \( L_B \) where \( P_B \) has productions \( S_B \rightarrow w_iS_Ba_i \) and \( S_B \rightarrow w_ia_i \)

**The union**

- The grammar
  \[ G = (\{S, S_B, S_A\}, \Sigma', P, S) \text{, where} \]
  \[ P = \{ S \rightarrow S_B, S \rightarrow S_A \} \cup P_A \cup P_B \]
  generates \( L_A \cup L_B \)
- The textbook shows that this \( G \) is ambiguous if and only if the PCP \((A, B)\) has a solution:

**ambiguity**

- The only source of ambiguity is
  \[ S \Rightarrow S_A \Rightarrow i_1w_{i_2} \ldots w_{i_m}a_{i_m}a_{i_{m-1}} \ldots a_{i_1} \] and
  \[ S \Rightarrow S_B \Rightarrow x_{i_1}x_{i_2} \ldots x_{i_m}a_{i_m}a_{i_{m-1}} \ldots a_{i_1} \]
- In which case, we have a solution of \((A, B)\):
  \[ i_1w_{i_2}w_{i_3} \ldots w_{i_m} = x_{i_1}x_{i_2}x_{i_3} \ldots x_{i_m} \]
- Thus, if ambiguity of a CFG were decidable then PCP would be decidable