Recitation 11 Notes

Context Free Grammars

Definition: A grammar $G = (V, T, P, S)$ is a context free grammar (cfg) if all productions in $P$ have the form $A \rightarrow x$

where

- $A \in V$, and
- $x \in (V \cup T)^*$.

Examples

Problem 1.
Given the language $L = \{a^n b^n : n \geq 0\}$, what is the CFG that generates the language?

$G = (\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow \epsilon\}, S)$

Problem 2.
Given the language $L = \{a^n b^k : k > n \geq 0\}$, what is the CFG that generates the language?

$G = (\{S, B\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow B, B \rightarrow bB, B \rightarrow b\}, S)$.

Problem 3.
The language $L = \{w : w \in \{a, b\}^*, n_a(w) = n_b(w)\}$, what is the CFG that defines the language?

$G = (\{S\}, \{a, b\}, \{S \rightarrow aSb, S \rightarrow bSa, S \rightarrow SS, S \rightarrow \epsilon\}, S)$

What is the PDA that recognizes this language?

1. Get input symbol
2. If input symbol is an ‘a’
   - If the stack is empty or stack_top = ‘A’,
     - push stack symbol ‘A’ on the stack
     - Go to Step 1.
   - If stack_top = ‘B’
     - Pop it from the stack
     - Go to Step 1.
3. If input symbol is a ‘b’
   - If the stack is empty or stack_top = ‘B’,
     - push stack symbol ‘B’ on the stack
- Go to Step 1.
  - If the stack_top = ‘A’
    - Pop it from the stack
    - Go to Step 1.
  4. If there are no more input symbols,
     - If stack is empty, then accept
     - Else Reject.

Problem 4.

The language L consists of a balanced set of parentheses. Write a CFG that generates the language.

\[ G = (\{S\}, \{(,\}, \{S \rightarrow (S), S \rightarrow SS, S \rightarrow \varepsilon\}, S). \]

**Definition:** A *sentential form* is the start symbol S of a grammar or any string in \((V \cup T)^*\) that can be derived from S.

Consider the grammar

\[
G = (\{S, A, B, C\}, \{a, b, c\}, P, S)
\]

where

\[
P = \{S \rightarrow ABC, A \rightarrow aA, A \rightarrow \varepsilon, B \rightarrow bB, B \rightarrow \varepsilon, C \rightarrow cC, C \rightarrow \varepsilon\}.
\]

**Derivation 1:**

\[
S \Rightarrow ABC \Rightarrow aABC \Rightarrow aABcC \Rightarrow aBcC \Rightarrow abBcC \Rightarrow abbc \Rightarrow abbc \Rightarrow abbc
\]

**Definition:** In a *leftmost derivation*, the leftmost variable is always expanded first:

\[
S \Rightarrow ABC \Rightarrow aABC \Rightarrow aBC \Rightarrow abBC \Rightarrow abbc \Rightarrow abbc \Rightarrow abbc
\]

**Definition:** In a *rightmost derivation*, the rightmost variable is always expanded first:

\[
S \Rightarrow ABC \Rightarrow ABcC \Rightarrow ABc \Rightarrow AbBc \Rightarrow Abbc \Rightarrow Abbc \Rightarrow aAbbc \Rightarrow abbc
\]
Derivation Trees

Consider Derivation 1, the tree can be constructed as follows:

The yield of the tree is the final string obtained by reading the leaves of the tree from left to right, ignoring the ε’s (unless all the leaves are ε, in which case the yield is ε). The yield of the above tree is the string abbc, as expected.

Ambiguity

Consider the grammar,

G = ({S}, {a, b}, {S→aSb, S→bSa, S→SS, S→ε}, S).

‘abab’ can be derived from the grammar with two parse trees.

Definition: A grammar G is ambiguous if there exists some string w ∈ L(G) for which

- there are two or more distinct derivation trees, or
- there are two or more distinct leftmost derivations, or
- there are two or more distinct rightmost derivations.

Definition: An inherently ambiguous language is a language for which no unambiguous grammar exists.
Push Down Automata

Definition: A nondeterministic pushdown automaton or NPDA is a 7-tuple
\[ M = (Q, \Sigma, \Gamma, \delta, q_0, z, F) \]
where
- \( Q \) is a finite set of states,
- \( \Sigma \) is the input alphabet,
- \( \Gamma \) is the stack alphabet,
- \( \delta \) is the transition function,
- \( q_0 \in Q \) is the initial state,
- \( z \in \Gamma \) is the stack start symbol, and
- \( F \subseteq Q \) is a set of final states.

The transition function \( \delta \) is defined as \( Q \times (\Sigma \cup \varepsilon) \times \Gamma \rightarrow Q \times \Gamma^* \)

Definition: An instantaneous description of a pushdown automaton is a triplet \((q, w, u)\), where
- \( q \) is the current state of the automaton,
- \( w \) is the unread part of the input string, and
- \( u \) is the stack contents (written as a string, with the leftmost symbol at the top of the stack).

Definition: A grammar is in Chomsky Normal Form if all productions are of the form
\[ A \rightarrow BC \]
or
\[ A \rightarrow a \]
Where
- \( A, B, \) and \( C \) are variables
- \( a \) is a terminal.

Definition: A grammar is in Greibach Normal Form if all productions are of the form
\[ A \rightarrow ax \]
Where
- \( a \) is a terminal
- \( x \in \mathcal{V}^* \).
Converting from CFG to NPDA

Idea: Any string of a context-free language has a leftmost derivation. So, set up the NPDA so that the stack contents "correspond" to this sentential form; every move of the NPDA represents one derivation step.

- Start state \( q_0 \) just gets things initialized. Create a transition from \( q_0 \) to \( q_1 \) to put the grammar's start symbol on the stack.
  \[
  \delta(q_0, \epsilon, z) \rightarrow \{(q_1, Sz)\}
  \]
- State \( q_1 \) does the bulk of the work, represent every derivation step as a move from \( q_1 \) to \( q_1 \).
- Use the transition from \( q_1 \) to \( q_f \) to accept the string.
  \[
  \delta(q_1, \epsilon, z) \rightarrow \{(q_f, z)\}
  \]

Example: Consider the grammar \( G = (\{S, A, B\}, \{a, b\}, P, S) \), where \( P = \{S \rightarrow a, S \rightarrow aAB, A \rightarrow aA, A \rightarrow a, B \rightarrow bB, B \rightarrow b\} \).

Rearrange the production \( S \rightarrow aAB \) as \( \delta(q_1, a, S) \rightarrow \{(q_1, AB)\} \)

<table>
<thead>
<tr>
<th>Initial Step</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow a )</td>
<td>( \delta(q_1, a, S) \rightarrow {(q_1, a)} )</td>
</tr>
<tr>
<td>( S \rightarrow aAB )</td>
<td>( \delta(q_1, a, S) \rightarrow {(q_1, AB)} )</td>
</tr>
<tr>
<td>( A \rightarrow aA )</td>
<td>( \delta(q_1, a, A) \rightarrow {(q_1, A)} )</td>
</tr>
<tr>
<td>( A \rightarrow a )</td>
<td>( \delta(q_1, a, A) \rightarrow {(q_1, \epsilon)} )</td>
</tr>
<tr>
<td>( B \rightarrow bB )</td>
<td>( \delta(q_1, a, B) \rightarrow {(q_1, B)} )</td>
</tr>
<tr>
<td>( B \rightarrow a )</td>
<td>( \delta(q_1, a, B) \rightarrow {(q_1, \epsilon)} )</td>
</tr>
<tr>
<td>Accept Step</td>
<td>( \delta(q_f, \epsilon, z) \rightarrow {(q_f, z)} )</td>
</tr>
</tbody>
</table>
For example, the derivation

$$S \Rightarrow Aab \Rightarrow aaB \Rightarrow aabB \Rightarrow aabb$$

maps into the sequence of moves

$$(q_0, aabb, z) \mid\rightarrow (q_1, aabb, Sz) \mid\rightarrow (q_1, abb, ABz) \mid\rightarrow (q_1, bb, Bz)$$

$$(q_1, b, Bz) \mid\rightarrow (q_1, \varepsilon, z) \mid\rightarrow (q_2, \varepsilon, \varepsilon)$$