Commonly used CFL

Arithmetic expressions

\[ S \rightarrow S + S \mid S - S \mid S \times S \mid S \div S \mid (S) \mid V \mid N \]

\[ V \rightarrow \{a, b, c, \ldots, z\} \cup V \mid \{a, b, c, \ldots, z\} \]

\[ N \rightarrow \{0, 1, 2, \ldots, 9\} \cup N \mid \{0, 1, 2, \ldots, 9\} \]

That produces expressions like

\[(x + 9)4y + 10\quad \text{etc.}\]

Balanced parentheses

\[ S \rightarrow (S) \mid S \cdot S \mid () \]

That produces

\[(())(())(()())()\quad \text{etc.}\]
Conversion of Arbitrary CFG to normal forms

<table>
<thead>
<tr>
<th>Chomsky Normal Form</th>
<th>Greibach Normal Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prod. of the form A → BC</td>
<td>A → a α</td>
</tr>
<tr>
<td>exactly 2 variables w one terminal on the Right hand side</td>
<td>α ∈ {ΣTUV}</td>
</tr>
<tr>
<td></td>
<td>i.e. it must start with a terminal</td>
</tr>
</tbody>
</table>

Steps of transformation

1. Eliminate useless symbols:
   - If a variable \( X \) doesn't appear in any derivation, i.e. \( S \xrightarrow{*} \alpha \beta X \alpha' \) or \( \beta X \xrightarrow{*} T \beta' \) \( X \) doesn't lead to any string over terminals, such variables (and all rules containing them) can be discarded without changing the language.

2. Eliminate \( \epsilon \) productions.
   - Suppose \( S \rightarrow A \beta \epsilon \) and \( A \rightarrow \epsilon \) \( B \rightarrow \epsilon \)
then we can eliminate $A \rightarrow E, B \rightarrow E$
and add the following rules in lieu

$S \rightarrow Ad \mid S \rightarrow Bd \mid S \rightarrow ABDd \mid S \rightarrow \alpha$

e.g., and all derivations of the original grammar can be preserved

(iii) Unit produc $A \rightarrow B, B \rightarrow \alpha \ldots$
can also be eliminated like

$A \rightarrow \times B, B \rightarrow \alpha$ then $A \rightarrow \alpha$

\[\text{Thm}: \text{Given any arbitrary CFG } G\]

it can be transformed to $G_1, G_2$
where $G_1$ is in CNF and $G_2$ in GNF.

$s.t. \quad L(G) = L(G_1) = L(G_2)$

We will show several applications of
the normal forms as they are
much easier to work with and have
nice properties.
\[ S \to aB \mid bA \]
\[ A \to a \mid aS \mid bAA \]
\[ B \to b \mid bS \mid aBB \]

\[ S \to CaB \mid C_bA \]
\[ A \to a \mid CaS \mid C_bD \]
\[ B \to b \mid C_bS \mid CaE \]
\[ C_a \to a \quad C_b \to b \]
\[ D \to AA \]
\[ E \to BB \]

**Membership problem** Given \( G = (V, \Sigma, \delta, \psi) \) in CNF and a string \( w \in \Sigma^* \) does \( S \xrightarrow{*} w \)?

\[ |w| = n \quad \omega_1, \omega_2, \omega_3 \ldots \omega_n \]
\[ S \xrightarrow{*} w \]

\[ S \xrightarrow{*} w \text{ iff there is a } j \text{ st. } \]
\[ S \to A \ B \text{ and } A \xrightarrow{*} \omega_1, \omega_2, \ldots, \omega_j \]
\[ \text{ and } B \xrightarrow{*} \omega_j, \omega_{j+1}, \ldots, \omega_n \]

\[ \omega_{ij} = \omega_i \omega_{i+1} \ldots \omega_j \]
1. **Start and Length Diagram**:
   - Variables: S, A, B, C_a, C_b
   - Production Rules:
     - S → C_a B | C_b A
     - A → a | C_a S | C_b D
     - B → b | C_b S | C_a E
     - D → AA
     - E → BB
     - C_a → a
     - C_b → b
   - Example: baab

2. **Grammar Table**:
   - | length 1 | 2 | 3 | 4 |
   - | S | B, C_b | A, C_a | A, C_a | B, C_b |
   - | S | D | S |
   - | 3 | A | A |
   - | 4 | S |

3. **Regular Expression**:
   - S → C_b A
   - C_b → b
   - A → aab
The D.P. takes $O(n^3)$ steps (considering the size of grammar to be constant and ignoring data structure cost)

CYK algorithm