CSC-257
Theory Of Computation
(BSc CSIT, TU)

Ganesh Khatri
kh6ganesh@gmail.com

Backus-Naur Form(BNF)

- Another way of representing CFG.
- It is named after John Backus, who invented it, and Peter Naur, who refined it.
- The traditional notation used by computer scientists to represent a context-free grammar
- Used to specify the syntactic rule of many computer languages, like Java
- Here, concept is similar to CFG, only the difference is instead of using symbol "→" in production, we use symbol ::=
- We enclose all non-terminals in angle brackets : <>
- An example of its use as a metalanguage would be in defining an arithmetic expression :

  <expr> ::= <term>|<expr><addop><term>

Backus-Naur Form(BNF)

- Example : The BNF for identifiers can be constructed as :

  <identifier> ::= <letter or underscore> | <identifier> | <symbol>

  <letter or underscore> ::= <letter> | <_>

  <symbol> ::= <letter or underscore> | <digit>

  <letter> ::= a | b | ……………….. | z

  <digit> ::= 0 | 1| 2 | ………………. | 9

Closure Property of CFL

- Here are some of the principal closure properties for context free languages

1. The context free language are closed under union

i.e. Given any two context free languages L_1 and L_2, their union $L_1 \cup L_2$ is also context free language

- Proof : Let $G_1 = (V_1, T_1, P_1, S_1)$ and $G_2 = (V_2, T_2, P_2$ and $S_2)$ be two context free grammars defining the languages $L(G_1)$ and $L(G_2)$.
- Without loss of generality, let us assume that they have common terminal set T, and disjoint set of non-terminals. Because, the non-terminals are distinct so the productions P_1 and P_2
- Let S be a new non-terminal not in V_1 and V_2. Then, construct a new grammar $G = (V, T, P, S)$ where :

$V = V_1 \cup V_2 \cup \{S\}$

$P = P_1 \cup P_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}$

Closure Property of CFL

1. The context free language are closed under union

- G is clearly a context free grammar because the two new productions so added are also of the correct form, we claim that L (G) = L (G_1) U L (G_2)
- For this, Suppose that $x \in$ L (G_1). Then there is a derivation of x as : $S_1 \rightarrow^* x$
- But in G, we have production, $S \rightarrow S_1$, so, there is a derivation of x also in G as :

  $S \rightarrow S_1 \rightarrow^* x$
- Thus, $x \in L(G)$. Therefore, $L(G_1) \subseteq L(G)$. A similar argument shows $L(G_2) \subseteq L(G)$
- So, we have, $L(G_1)$ U $L(G_2) \subseteq L(G)$ [$\subseteq$ **: is subset of**]

Closure Property of CFL

1. The context free language are closed under union

- Conversely, suppose that $x \in L(G)$. Then there is a derivation of x in G as :

  $S \rightarrow \beta \rightarrow^* x$

- Because of the way in which P is constructed, β must be either S_1 or S_2
- Suppose $\beta = S_1$. Any derivation in G of the form $S_1 \rightarrow^* x$ must involve only productions of G_1 so, $S_1 \rightarrow^* x$ is a derivation of x in G_1.
- Hence, $\beta = S_1 \rightarrow x \in L(G_1)$
- Thus L(G) is subset of $L(G_1)$ U $L(G_2)$
- It follows that L(G) = L(G1) U L(G2)

Closure Property of CFL

- Similarly following two closure properties of CFL can be proved as for the union we have proved.

  2. **The CFLs are closed under concatenation**
  3. **The CFLs are closed under Kleene closure**