BITS Pilani
Pilani Campus

Regular Grammar, Derivation and Ambiguity

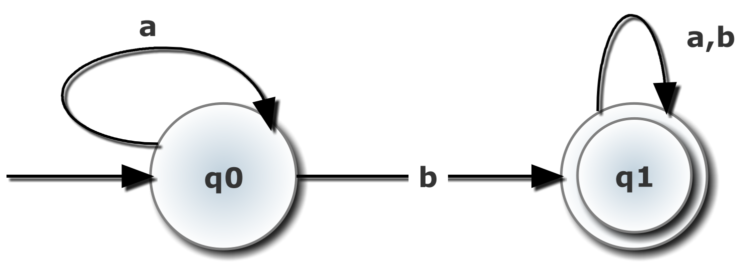
Shashank Gupta
Assistant Professor
Department of Computer Science and Information Systems

Equivalence between DFA and Regular Grammar

Let L be a regular language accepted via a DFA D = $(Q, \Sigma, \delta, q_0, F)$. We will construct a Regular Grammar G = (V, Σ, P, R_0) for L.

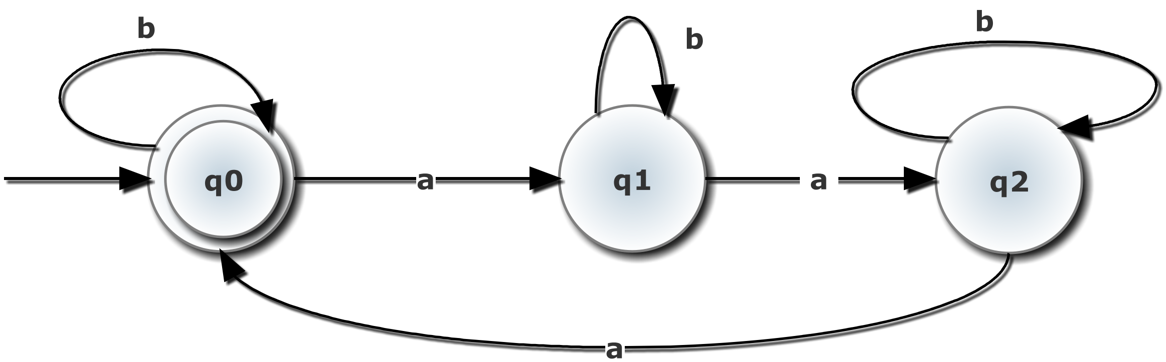
- **V** = $\{R_i \mid q_i \in Q, \forall i\}$. In other words, there is a variable in the G corresponding to every state in the DFA.
- The set of terminals is the same as the alphabet of D.
- If $\delta(q_i, a) = q_j$, then the substitution rule $R_i \rightarrow aR_j$ is in **P**.
- Also if $q_i \in F$, then $R_i \rightarrow \in$ is a substitution rule in **P**.
- $\mathbf{R_0}$ is the start variable, that is, the variable corresponding to the start state.

Examples



$$A \rightarrow aA \mid bB$$
$$B \rightarrow aB \mid bB \mid \epsilon$$

More Examples



$$S \rightarrow bS \mid aA \mid \in$$
$$A \rightarrow aB \mid bA$$
$$B \rightarrow aS \mid bB$$

Derivation and Ambiguity in CFG

Derivation

It is the process of deriving the strings from the Given Grammar.

- By replacing the variables with the terminal symbols.
- In each step of substitution, we get the sentential form of the Grammar.

However, if there are multiple variables on the RHS of production rule then

- Which variable must be choose for replacement.

Types of Derivation

Leftmost Derivation

- Replace only the leftmost non-terminal by some production rule at each step.

Rightmost Derivation

- Replace only the rightmost non-terminal by some production rule at each step.

Parse Trees

A parse tree for a string w with respect to a CFG G is a rooted, ordered tree that represents the derivation of w with respect to the grammar.

It represents the "syntactic" structure of the string.

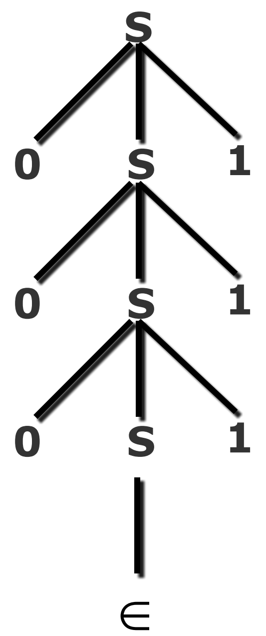
- Note that, in a parse tree concatenating the symbols from left to right gives the original string.

Example

Consider the following CFG, G1

$S \rightarrow 0S1 \mid \in$

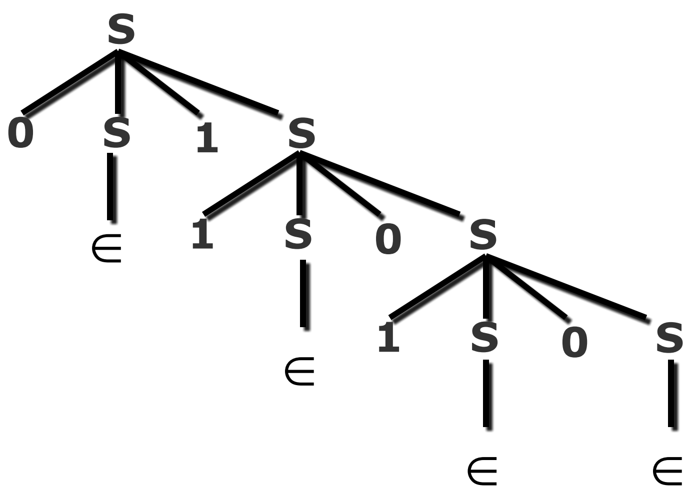
Parse Tree for the string 000111



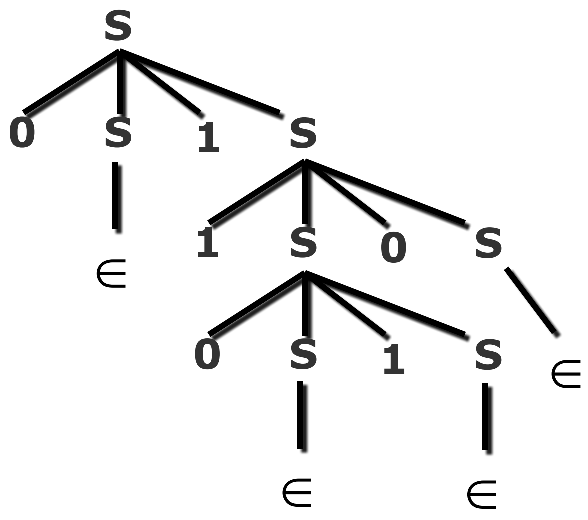
More Examples

Consider the following CFG, G2

$S \rightarrow 1S0S \mid 0S1S \mid \in$



Parse Tree for the string 011010



Different Parse Tree for the same string 011010

Properties of Parse Tree

Every internal node of the tree is labelled by a variable.

Every leaf node is labelled either by a terminal or by $\in$. Moreover, if a leaf is labelled $\in$ then, it is the only child of its parent.

- If an internal node is labelled A and its children are labelled X_1, X_2, . . . , X_k in order from left to right, then $A \rightarrow X_1X_2 \ldots X_k$ is a rule in the CFG.

Derivation and Ambiguity

The derivation of a string w with respect to a CFG is a sequence of substitutions that yields w starting from the start variable.

- Derivation of 000111 with respect to G1: **S-> 0S1** | $\in$
- $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000S111 \Rightarrow 000111$
- Observe that the string 000111 has only one derivation with respect to G1, because the string generated in each intermediate step has **exactly one variable on the RHS**. Every string in L(G1) has exactly one derivation.

Derivation and Ambiguity (Continued…..)

G2: $S \to 1S0S \mid 0S1S \mid \epsilon$

Consider the string 0101 in the language of the CFG G2. Below we show 3 derivations of the string with respect to G2.

- **Derivation 1**: $S \Rightarrow 0S1S \Rightarrow 01S \Rightarrow 010S1S \Rightarrow 0101S \Rightarrow 0101$
- **Derivation 2**: $S \Rightarrow 0S1S \Rightarrow 0S10S1S \Rightarrow 0S101S \Rightarrow 0S101 \Rightarrow 0101$
- **Derivation 3**: $S \Rightarrow 0S1S \Rightarrow 01S0S1S \Rightarrow 010S1S \Rightarrow 0101S \Rightarrow 0101$

In the above example above, derivations 1 and 3 are leftmost derivations, whereas 2 is not.

- We can say that w is ambiguous with respect to G if w has at least 2 leftmost derivations with respect to G. Otherwise we say that w is unambiguous.

Derivation and Ambiguity (Continued…..)

G1: $S \rightarrow 0S1 \mid \epsilon$

G2: $S \rightarrow 1S0S \mid 0S1S \mid \epsilon$

A CFG G is said to be ambiguous if it generates some string ambiguously.

- The grammar G1 above is unambiguous and G2 is ambiguous.

A CFL is said to be inherently ambiguous if all CFGs that accept the language are ambiguous

- Let G be a grammar and w $\in$ L(G). w is unambiguous with respect to G if and only if w has a **unique parse tree** with respect to G..

Examples

Check whether the following grammar G1 and G2 are ambiguous or not

G1: $S \rightarrow SS \mid a$

G2: $S \rightarrow aSb \mid SS \mid \in$