Formal Languages
Grammar and Types

N Geetha
AM & CS
PSG Tech

Formal Language

- Considers Language as a mathematical object
- An alphabet is a finite set of symbols.
- Examples
- $\Sigma_1 = \{a, b, c, d, \ldots, z\}$: the set of letters in English
- $\Sigma_2 = \{0, 1, \ldots, 9\}$: the set of (base 10) digits
- $\Sigma_3 = \{a, b, \ldots, z, \#\}$: the set of letters plus $\#$
- $\Sigma_4 = \{\ (,\ )\ \}$: the set of open and closed brackets
- A language is a set of strings over an alphabet.

String

- A string over alphabet Σ is a finite sequence of symbols in Σ.
- The empty string will be denoted by ε
- Examples
- abfbz is a string over $\Sigma_1 = \{\mathrm{a, b, c, d, \ldots, z}\}$
- 9021 is a string over $\Sigma_2 = \{0, 1, \ldots, 9\}$
- $\mathrm{ab\#bc}$ is a string over $\Sigma_3 = \{\mathrm{a, b, \ldots, z, \#}\}$
- $))()(()$ is a string over $\Sigma_4 = \{\ (\ ,\ )\ \}$

Language

- L *over* V *is any subset of* V^* $L \subseteq V^*$
- $L_1 =$ The set of all strings over $\Sigma_1 = \{a, b, c, d, \ldots, z\}$ that contain the substring "fool"
- $L_2 =$ The set of all strings over $\Sigma_2 = \{0, 1, \ldots, 9\}$ that are divisible by 7 $= \{7, 14, 21, \ldots\}$
- $L_3 =$ The set of all strings of the form s#s where s is any string over $\{a, b, \ldots, z\}$
- $L_4 =$ The set of all strings over $\Sigma_4 = \{ (,) \}$ where every (can be matched with a subsequent)

Grammar

- Is related to studies in natural languages
- Concerned with
  - Defining valid sentences of a language
  - Providing a structural definition of such valid sentences
- Provides a set of rules by which all valid strings can be generated

Formal Grammar

- Introduced by the linguist Noam Chomsky in 1950s, now a Professor of Emeritus at MIT
- A Grammar is a 4-tuple G = (N, T, S,P) where
- N : set of finite non-terminal symbols
- T : set of finite terminal symbols
- S : $S \in N$ is the start symbol
- P : finite set of production rules : $\{\alpha \rightarrow \beta$ / α, β are combinations of N and T $\}$

Notion of derivation

- To characterize a Language starting from a Grammar we need to introduce the notion of Derivation.
- The notion of Derivation uses Productions to generate a string starting from another string.
- **Direct Derivation (in symbols $\Rightarrow$).**
  If $\alpha \rightarrow \beta \in P$ and $\gamma, \delta \in V^*$, then, $\gamma\alpha\delta \Rightarrow \gamma\beta\delta$.
- **Derivation (in symbols $\Rightarrow^*$).**
  If $\alpha_1 \Rightarrow \alpha_2, \alpha_2 \Rightarrow \alpha_3, \ldots, \alpha_{n-1} \Rightarrow \alpha_n$, then, $\alpha_1 \Rightarrow^* \alpha_n$.

Language of Grammar

Let $G = (V, T, S, P)$ be a phrase-structure grammar. The *language generated by G* (or the *language of G*), denoted by $L(G)$, is the set of all strings of terminals that are derivable from the starting state S. In other words,

$$L(G) = \{w \in T^* \mid S \stackrel{*}{\Rightarrow} w\}.$$

Let G be the grammar with vocabulary $V = \{S, A, a, b\}$, set of terminals $T = \{a, b\}$, starting symbol S, and productions $P = \{S \to aA, S \to b, A \to aa\}$. What is $L(G)$, the language of this grammar?

Let G be the grammar with vocabulary $V = \{S, 0, 1\}$, set of terminals $T = \{0, 1\}$, starting symbol S, and productions $P = \{S \to 11S, S \to 0\}$. What is $L(G)$, the language of this grammar?

Word of L(G)

Determine whether the word *cbab* belongs to the language generated by the grammar $G = (V, T, S, P)$, where $V = \{a, b, c, A, B, C, S\}$, $T = \{a, b, c\}$, S is the starting symbol, and the productions are

$$\begin{aligned}
S &\to AB \\
A &\to Ca \\
B &\to Ba \\
B &\to Cb \\
B &\to b \\
C &\to cb \\
C &\to b.
\end{aligned}$$

Example

Example 1. Let us consider the following Grammar, G= (V_T,V_N,S,P):

- $V_T = \{0, 1\}$;
- $V_N = \{S\}$;
- $P = \{S \to 0S1, S \to \epsilon\}$;

Then:

- $S \Rightarrow^* 0^n1^n$;
- $L(G) = \{0^n1^n \mid n \geq 0\}$.

Example

Example 2. Let us consider the following Grammar, G= (V_T,V_N,S,P):

- $V_T = \{a, b\}$;
- $V_N = \{S, A, B\}$;
- $S = S$.

With Productions in P:

$$
\begin{array}{llll}
r1. & S & \rightarrow & AB \\
r2. & A & \rightarrow & aA \\
r3. & A & \rightarrow & \epsilon \\
r4. & B & \rightarrow & bB \\
r5. & B & \rightarrow & \epsilon
\end{array}
$$

Then:

- $$S \Rightarrow^{r1} AB \Rightarrow^{r2} aAB \Rightarrow^{r2} aaAB \Rightarrow^{r2} aaaAB \Rightarrow^{r3} aaaB \Rightarrow^{r4} aaabB \Rightarrow^{r4} aaabbB \Rightarrow^{r5} aaabb$$
- $L(G) = \{a^m b^n \mid m, n \geq 0\}$

Example

Example 3. Let us consider the following Grammar with more than one symbol on the left side of Productions, G= (V_T,V_N,S,P):

- $V_T = \{a\}$;
- $V_N = \{S, N, Q, R\}$;
- $S = S$.

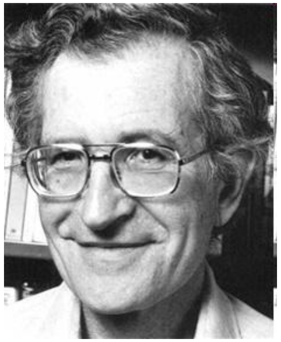
With Productions in P:

$r1.$	S	$\rightarrow$	QNQ
$r2.$	QN	$\rightarrow$	QR
$r3.$	RN	$\rightarrow$	NNR
$r4.$	RQ	$\rightarrow$	NNQ
$r5.$	N	$\rightarrow$	a
$r6.$	Q	$\rightarrow$	ϵ

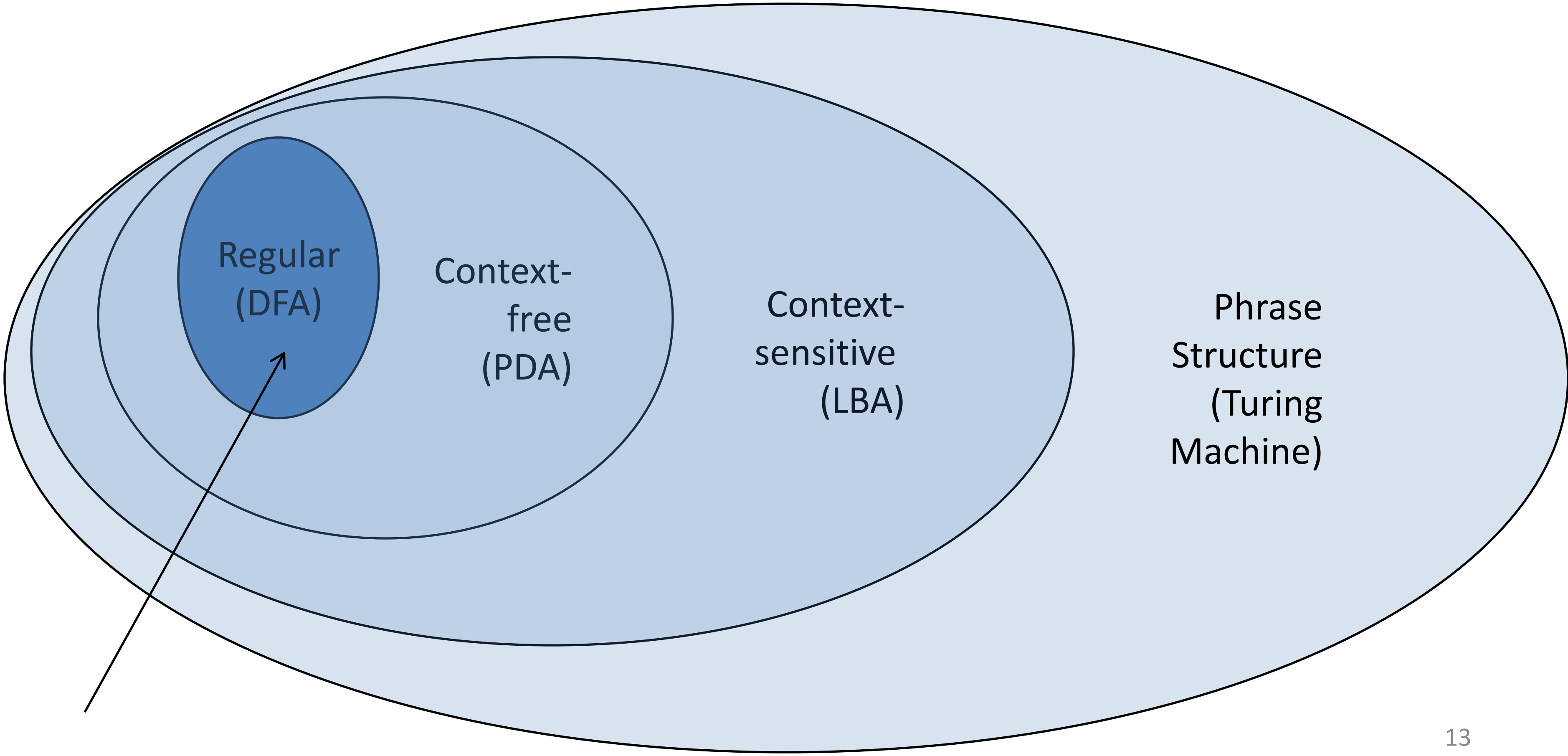
Then:

- $S \Rightarrow^{r1} QNQ \Rightarrow^{r2} QRQ \Rightarrow^{r4}$
  $QNNQ \Rightarrow^{r2} QRNQ \Rightarrow^{r3}$
  $QNNRQ \Rightarrow^{r4} QNNNNQ \Rightarrow^{*}$
  $aaaa$
- $L(G) = \{a^{(2^n)} \mid n \geq 0\}$

The Chomsky Hierachy



- A containment hierarchy of classes of formal languages



Chomsky Hierarchy

Chomsky Language Class	Grammar	Recognizer
3	Regular	Finite-State Automaton
2	Context-Free	Push-Down Automaton
1	Context-Sensitive	Linear-Bounded Automaton
0	Unrestricted	Turing Machine

Table 2.1. The Chomsky Hierarchy of languages and automata.

Type 3 or Regular Grammar (RG)

- G = (N,T,S,P); N = {A,B,S}; T = {a}
- Can have rules as left hand side single NonTerminal and RHS is a Terminal or Terminal followed by a NonTerminal; N -> T or N -> TN

Type 3. $A \rightarrow aB$, or $A \rightarrow a$

Furthermore, a rule of the following form is allowed:

$S \rightarrow \epsilon$

if S does not appear on the right side of any rule.

- The above define the *Right-Regular Grammars*. The following Productions:

  $A \rightarrow Ba$, or $A \rightarrow a$

  define *Left-Regular Grammars*.
- Right-Regular and Left-Regular Grammars define the same set of Languages.

Type 2 or Context Free Grammar(CFG)

- Can have productions only of the form u -> v where u is a single non-terminal and v is (NUT)*

Type 2. $A \rightarrow \beta$
with $A \in V_N$ and $\beta \in V^*$.

- The term "Context-Free" comes from the fact that the non-terminal A can always be replaced by β, in no matter what context it occurs.
- Context-Free Grammars are important because they are powerful enough to describe the syntax of programming languages; in fact, almost all programming languages are defined via Context-Free Grammars.

Type 1 or Context Sensitive Grammar (CSG)

- V = N U T

Type 1. $\alpha A \gamma \rightarrow \alpha \beta \gamma$
with $\alpha, \gamma \in V^*$, $\beta \in V^+$ and $A \in V_N$.
Furthermore, a rule of the following form is allowed:
$S \rightarrow \epsilon$
if S does not appear on the right side of any rule.

- A in the context of α and γ can be replaced by β
- Also know as length-increasing or non-contracting Grammar

Type 0 or Phrase Structure Grammar(PSG)

- No restriction in the productions of the grammar

Type 0. $\alpha \rightarrow \beta$
with $\alpha \in \mathsf{V}^* \cdot \mathsf{V}_\mathsf{N} \cdot \mathsf{V}^*$ and $\beta \in \mathsf{V}^*$.

Example for RG

- G = (N,T,S,P); N = {S, A}; T = {a,b},
- P = {S->aS, S -> aA, A->b}

Example for CFG

Example 1. Let us consider the following Grammar, G= (V_T,V_N,S,P):

- $V_T = \{0, 1\}$;
- $V_N = \{S\}$;
- $P = \{S \to 0S1, S \to \epsilon\}$;

Then:

- $S \Rightarrow^* 0^n1^n$;
- $L(G) = \{0^n1^n \mid n \geq 0\}$.

Example for CFG

Example 2. Let us consider the following Grammar, G= (V_T,V_N,S,P):

- $V_T = \{a, b\}$;
- $V_N = \{S, A, B\}$;
- $S = S$.

With Productions in P:

$$
\begin{array}{llll}
r1. & S & \rightarrow & AB \\
r2. & A & \rightarrow & aA \\
r3. & A & \rightarrow & \epsilon \\
r4. & B & \rightarrow & bB \\
r5. & B & \rightarrow & \epsilon
\end{array}
$$

Then:

- $$S \Rightarrow^{r1} AB \Rightarrow^{r2} aAB \Rightarrow^{r2} aaAB \Rightarrow^{r2} aaaAB \Rightarrow^{r3} aaaB \Rightarrow^{r4} aaabB \Rightarrow^{r4} aaabbB \Rightarrow^{r5} aaabb$$
- $L(G) = \{a^m b^n \mid m, n \geq 0\}$

Example for CSG

- G = ({S,A,B},{a,b}, S, {S ->aAB, AB -> bB, B ->b, B ->aB})

Example for PSG

Example 3. Let us consider the following Grammar with more than one symbol on the left side of Productions, G= (V_T,V_N,S,P):

- $V_T = \{a\}$;
- $V_N = \{S, N, Q, R\}$;
- $S = S$.

With Productions in P:

$$
\begin{array}{llcl}
r1. & S & \rightarrow & QNQ \\
r2. & QN & \rightarrow & QR \\
r3. & RN & \rightarrow & NNR \\
r4. & RQ & \rightarrow & NNQ \\
r5. & N & \rightarrow & a \\
r6. & Q & \rightarrow & \epsilon
\end{array}
$$

Then:

- $S \Rightarrow^{r1} QNQ \Rightarrow^{r2} QRQ \Rightarrow^{r4} QNNQ \Rightarrow^{r2} QRNQ \Rightarrow^{r3} QNNRQ \Rightarrow^{r4} QNNNNQ \Rightarrow^{*} aaaa$
- $L(G) = \{a^{(2^n)} \mid n \geq 0\}$