Specifying languages by grammars

Definition: *A **grammar** is an ordered quadruple* $G = (N, T, S, P)$*,*
Where:

*N is the alphabet of **variables** (or **nonterminal symbols**),*
*T is the alphabet of **terminal symbols**, where* $N \cap T = \emptyset$*,*
$S \in N$ *is the **start symbol**.*
P is a finite set of productions, each production of the following form:

$$u \rightarrow v$$

Such that:

$$u \in (N \cup T)^{*} N (N \cup T)^{*}$$
$$v \in (N \cup T)^{*}$$

***u** is called a left-hand side (L.H.S) of the production, and*
***v** is called the right-hand side (R.H.S.) of the production.*

If for a grammar there are more than one production with the same left-hand side, then this production: $u \rightarrow v_1, u \rightarrow v_2, \ldots\ldots. u \rightarrow v_r$

can be written as:

$$u \rightarrow v_1 \mid v_2 \mid \ldots\ldots v_r$$

Example:

G = ({A, B}, {0, 1, #}, A, P)
Where P is,

A → 0A1
A → B
B → #

P can be written as follows:

A → 0A1 | B
B → #

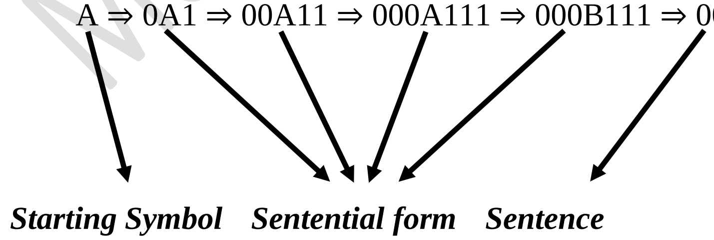
Derivation

The sequence of substitutions to generate a string (sentence) is called a **derivation**.

Example 1. A derivation of string 000#111 in G is

A ⇒ 0A1 ⇒ 00A11 ⇒ 000A111 ⇒ 000B111 ⇒ 000#111

G:
A → 0A1 | B
B → #



A **sentential form** is any string derivable from the start symbol.
A **sentence** is a **sentential form** consisting only of terminals

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

Example 2. Write a grammar for the following languages:

$L_1 = \{a^n b^n \mid n >= 0\}$

Answer:

G_1: $(\{A\}, \{a, b\}, A, P_1)$

Where P_1 is,

$A \rightarrow aAb \mid \epsilon$

$L_2 = \{a^n b^m \mid n, m >= 1\}$

Answer:

G_2: $(\{S, A, B\}, \{a, b\}, S, P_2)$

Where P_2 is,

$S \rightarrow AB$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

L_3 = all words over {a, b, c} such that each word starts and ends with a.

Answer:

G_3: $(\{S, A\}, \{a, b, c\}, S, P_3)$

Where P_3 is,

$S \rightarrow aAa$

$A \rightarrow aA \mid bA \mid cA \mid \epsilon$

e.g., acbaa $(S \Rightarrow aAa \Rightarrow acAa \Rightarrow acbAa \Rightarrow acbaAa \Rightarrow acbaa)$

L_4 = all words over {0, 1} such that each word contains an even number of 1's.

Answer:

G_4: $(\{E, O\}, \{0, 1\}, E, P_4\}$

Where P_4 is,

$E \rightarrow 1O \mid 0E \mid \epsilon$

$O \rightarrow 1E \mid 0O$

e.g., 01111 $(E \Rightarrow 0E \Rightarrow 01O \Rightarrow 011E \Rightarrow 0111O \Rightarrow 01111E \Rightarrow 01111)$

L_5 = all words over {0, 1} such that each word contains an odd number of 0's.

Answer:

G_5: $(\{E, O\}, \{0, 1\}, E, P_5\}$

Where P_5 is,

$E \rightarrow 0O \mid 1E$

$O \rightarrow 0E \mid 1O \mid \epsilon$

e.g., ***HW***

L_6 = all words over {0, 1} such that each word contains an even number of 1's and odd number of 0's.

Answer:

- Even Even e.g., (0110, 1001, 00, 11}
- **Even Odd e.g., (110, 01010, 0, 01100)**
- Odd Even e.g., (100, 10011, 1101101, 11100, 1)
- Odd Odd e.g., (10, 1101,001011)

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

G_6= ({S_{ee}, S_{eo}, S_{oe}, S_{oo}}, {0, 1}, S_{ee}, P_6}
Where P_6 is,
See → 1Soe | 0Seo
Seo → 1Soo | 0See | ϵ
Soe → 1See | 0Soo
Soo → 1Seo | 0Soe

$L_7 = \{w \in \{a, b, c\}^* : w = a^n b^n c^n, n \geq 0 \}$
Answer:
G_7: ({S, B, C,}, {a, b, c}, S, P_7})
Where P_7 is,
S → aSBC | ε
CB → BC
aB → ab
bB → bb
bC → bc
cC → cc

Here is a sample derivation:
S ⇒ aSBC ⇒ aaSBCBC ⇒ aaaSBCBCBC ⇒ aaaBCBCBC ⇒ aaaBBCCBC ⇒ aaaBBCBCC ⇒ aaaBBBCCC ⇒ aaabBBCCC ⇒ aaabbBCCC ⇒ aaabbbCCC ⇒ aaabbbcCC ⇒ aaabbbccC ⇒ aaabbbccc

From the start, we generate a sentential form: $\mathbf{a^n (BC)^n}$
The rule CB → BC gives you the ability to change this to $\mathbf{a^n B^n C^n}$
The remaining rules allow you to convert this form to: $\mathbf{a^n b^n c^n}$

HW Find the grammar corresponding to the following Context Sensitive Languages:

1. $L = a^{2i} b^{3i} a^{2i} \mid i>=1$
2. $L = a^i b^i c^i d^i \mid i>=1$

The Chomsky hierarchy of languages
Chomsky Hierarchy (which was first formed in 1956) represents the class of languages that are accepted by a different machine. The category of language in Chomsky's Hierarchy is as given below:

***Type-0 grammar**:*
A grammar G is of type 0 **(unrestricted grammar)** if there are no restrictions on the productions. Type-0 grammar is also known as an a **phrase-structure grammar**. This grammar generates exactly all languages that can be efficiently modelled/recognised by *a Turing machine*.

***Type-1 grammar**:*
A grammar G is of type 1 **(context-sensitive grammar)** if all of its productions are of the form $\alpha A\gamma \rightarrow \alpha\beta\gamma$, where $A \in N$ (i.e., A is a single nonterminal), $\alpha, \gamma \in (N \cup T)^*$ (i.e., α and γ are strings of nonterminals and terminals), $\beta \in (N \cup T)^+$ (i.e., β is a nonempty string of nonterminals and terminals). A production of the form S → ε can also be accepted if the start symbol S does not occur in the right-hand side of any production. *Context-sensitive grammars generate the context-sensitive languages*.

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

The languages described by these grammars are exactly all languages that can be recognised by a *non-deterministic Turing machine* whose tape is bounded by constant times the length of the input.

Type-2 grammar:

A grammar G is of type 2 **(context-free grammar)** if all of its productions are of the form $A \rightarrow \beta$, where $A \in N$, $\beta \in (N \cup T)^+$. A production of the form $S \rightarrow \varepsilon$ can also be accepted if the start symbol S does not occur in the R.H.S. of any production. *Context-free grammars generate the Context-free languages*. These languages are exactly all languages that can be recognized by a *non-deterministic pushdown automaton*. Context free languages are the theoretical basis for the **syntax** of most programming languages.

أي أن طرف اليسار لهذه القواعد عبارة عن متحول وحيد وهو حرف كبير وقد يكون طرف اليمين هو تعاقب من رموز نهائية ولا نهائية (أحرف كبيرة وصغيرة).

Type-3 grammar:

A grammar G is of type 3 **(regular grammar)** if its productions are of the form $A \rightarrow aB$ or $A \rightarrow a$, where $a \in T$ and $A, B \in N$. A production of the form $S \rightarrow \varepsilon$ can also be accepted if the start symbol S does not occur in the R.H.S. of any production. *Regular grammar generates Regular languages* which are commonly used to define search patterns and the **lexical** structure of programming languages.

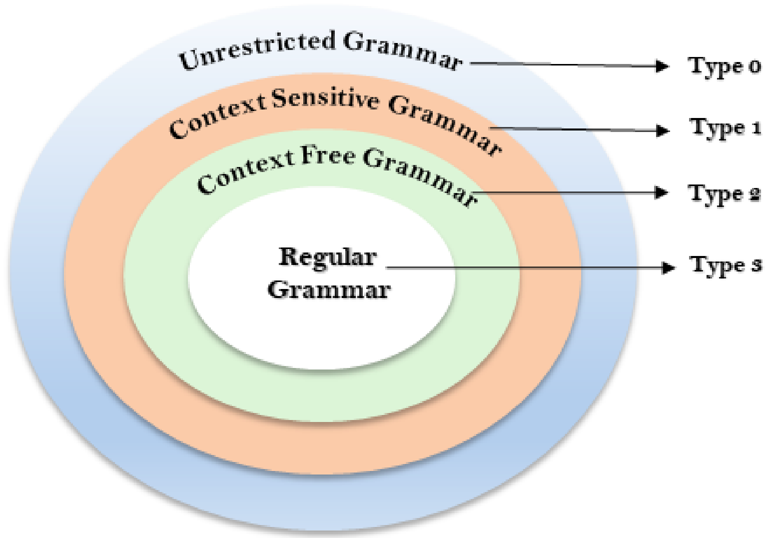
Note: *If a grammar G is of type i, then language L(G) is also of type i.*

This table summarizes each of Chomsky's four types of grammars, the class of language it generates, the type of automaton that recognizes it, and the form its rules must have:

GRAMMAR	LANGUAGES	AUTOMATA MACHINE	GRAMMAR RULE
Type 0: Unrestricted	Recursively Enumerable	Turing Machine	$bAa \rightarrow aa$, $S \rightarrow s$
Type 1: Context Sensitive	Context Sensitive	Linear Bounded Automaton	$aAb \rightarrow ayb$
Type 2: Context Free	Context Free	Pushdown Automaton	$A \rightarrow aBb$
Type 3: Regular	Regular	Finite State Automaton	$A \rightarrow a$ and $A \rightarrow aB$

Every regular language is context-free, every context-free language is context-sensitive, every context-sensitive language is recursive, and every recursive language is recursively enumerable. This figure shows set inclusions described by the Chomsky hierarchy:

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.



Example: what is the type of each of the following grammar:

1.

$G_1 = (N_1, T_1, P_1, S_1)$, where $N_1 = \{S_1, A, B, C\}$, $T_1 = \{a, 0, 1\}$.

Elements of P_1 are:

$S_1 \rightarrow ACA$		Type 2	
$AC \rightarrow AACA \mid ABa \mid AaB$	$€^{\alpha} A^{A} C^{\gamma} \rightarrow €^{\alpha} \underline{AAC}^{\beta} A^{\gamma}$	-	
	$€^{\alpha} A^{A} C^{\gamma} \rightarrow €^{\alpha} \underline{AACA}^{\beta} €^{\gamma}$	-	
	$A^{\alpha} C^{A} €^{\gamma} \rightarrow A^{\alpha} \underline{AACA}^{\beta} €^{\gamma}$	Type 1	Type 1
	$A^{\alpha} C^{A} €^{\gamma} \rightarrow A^{\alpha} \underline{ABa}^{\beta} €^{\gamma}$	Type 1	
	$A^{\alpha} C^{A} €^{\gamma} \rightarrow A^{\alpha} \underline{AaB}^{\beta} €^{\gamma}$	Type 1	
$B \rightarrow AB \mid A$		Type 2	
$A \rightarrow 0 \mid 1$		Type 3	

2.

$G_2 = (N_2, T_2, P_2, S)$, where $N_2 = \{S, A, B\}$, $T_2 = \{a, +, *, (,)\}$.

Elements of p_2 are:

$S \rightarrow S+A \mid A$	Type 2	
$A \rightarrow A*B \mid B$	Type 2	Type 2
$B \rightarrow (S) \mid a$	Type 2	

3.

$G_3 = (N_3, T_3, P_3, S)$, where $N_3 = \{S, A, B\}$, $T_3 = \{a, b\}$.

Elements of p_3 are:

$S \rightarrow aA$	Type 3	
$A \rightarrow aB \mid a$	Type 3	Type 3
$B \rightarrow aB \mid bB \mid a \mid b$	Type 3	

4.

$aS \rightarrow SAa \mid aA$	$a^{\alpha} S^{A} €^{\gamma} \rightarrow S^{\alpha} \underline{Aa}^{\beta} €^{\gamma}$	Type 0	
	$a^{\alpha} S^{A} €^{\gamma} \rightarrow a^{\alpha} \underline{A}^{\beta} €^{\gamma}$	Type 1	Type 0
$aA \rightarrow abc$	$a^{\alpha} S^{A} €^{\gamma} \rightarrow a^{\alpha} \underline{bc}^{\beta} €^{\gamma}$	Type 1	

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.