Regulated Re-writing

In a given grammar, re-writing can take place at a step of a derivation by the usage of any applicable rule in any desired place. That is if A is a nonterminal occurring in any sentential form say $\alpha A\beta$, the rules being

$$A \rightarrow \gamma$$

$$A \rightarrow \delta$$

then any of these two rules are applicable for the occurrence of A in $\alpha A\beta$. Hence, one encounters nondeterminism in its application. One way of naturally restricting the nondeterminism is by regulating devices, which can select only certain derivations as correct in such a way that the obtained language has certain useful properties. For example, a very simple and natural control on regular rules may yield a non-regular language.
While defining the four types of grammars, we put restrictions in the form of production rules to go from type 0 to type 1, then to type 2 and type 3. In this chapter we put restrictions on the manner of applying the rules and study the effect. There are several methods to control re-writing, some of the standard control strategies are as follows.
Matrix Grammar

A matrix grammar is a quadruple $G = (N, T, P, S)$ where $N$, $T$ and $S$ are as in any Chomsky grammar. $P$ is a finite set of sequences of the form:

$$m = [\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \ldots, \alpha_n \rightarrow \beta_n]$$

$n \geq 1$, with $\alpha_i \in (N \cup T)^+$, $\beta_i \in (N \cup T)^*$, $1 \leq i \leq n$. $m$ is a member of $P$ and a ‘matrix’ of $P$.

$G$ is a matrix grammar of type $i$, where $i \in \{0,1,2,3\}$, if and only if the grammar $G_m = (N, T, m, S)$ is of type $i$ for every $m \in P$.

Similarly, $G$ is $\varepsilon$–free if each $G_m$ is $\varepsilon$–free.
Definition 1

Let \( G = (N, T, P, S) \) be a matrix grammar. For any two strings \( u, v \in (N \cup T)^+ \), we write \( u \xrightarrow{G} v \) (or \( u \Rightarrow v \) if there is no confusion on \( G \)), if and only if there are strings \( u_0, u_1, u_2, \ldots, u_n \) in \((N \cup T)^+\) and a matrix \( m \in M \) such that \( u = u_0, u_n = v \) and

\[
\begin{align*}
  u_{i-1} &= u_{i-1}x_iu_i'' \\
  u_i &= u_{i-1}y_iu_{i-1}
\end{align*}
\]

for some \( u_{i-1}, u_i'' \) for all \( 0 \leq i \leq n-1 \) and \( x_i \to y_i \in m, 1 \leq i \leq n \).

Clearly, any direct derivation in a matrix grammar \( G \) corresponds to an \( n \)-step derivation by \( G_m = (N, T, P, S) \). That is, the rules in \( m \) are used in sequence to reach \( v \). \( \xrightarrow{*} \) is the reflexive, transitive closure of \( \Rightarrow \) and

\[
L(G) = \{ w/ w \in T^*, S \Rightarrow^* w \}.
\]
Definition 2

Let $G = (N, T, P, S)$ be a matrix grammar. Let $F$ be a subset of rules of $M$. We now use the rules of $F$ such that, the rules in $F$ can be passed over if they cannot be applied, whereas the other rules in any matrix $m \in P$ not in $F$ must be used. That is, for

$$u, v \in (N \cup T)^+, u \xrightarrow{m} v,$$

if and only if there are strings $u_0, u_1, \ldots, u_n$ and a matrix $m \in M$ with rules $\{r_1, r_2, \ldots, r_n\}$ (say), with $r_i$:

$$x_i \rightarrow y_i \quad 1 \leq i \leq n.$$

Then, either $u_{i-1} = u_{i-1}^\prime x_i u_{i-1}^{\prime\prime}$, $u_i = u_{i-1}^\prime y_i u_{i-1}^{\prime\prime}$ or the rule $x_i \rightarrow y_i \in F$. Then $u_i = u_{i-1}$.
This restriction by $F$ on any derivation is denoted as $\Rightarrow_{ac}$, where ‘ac’ stands for ‘appearance checking’ derivation mode. Then,

$$L(G,F) = \left\{ w / S \Rightarrow_{ac} w, w \in T^* \right\}$$

Let $M(M_{ac})$ denote the family of matrix languages without appearance checking (with appearance checking) of type 2 without $\varepsilon$-rules.

Let $M^\lambda(M_{ac}^\lambda)$ denote the family of matrix languages without appearance checking (with appearance checking) of type 2 with $\varepsilon$-rules.
Example 1

Let \( G = (N, T, P, S) \) be a matrix grammar where

\( N = \{S, A, B, C, D\} \)

\( T = \{a, b, c, d\} \)

\( P = \{P_1, P_2, P_3, P_4\} \), where

\( P_1 : [S \rightarrow ABCD] \)

\( P_2 : [A \rightarrow aA, B \rightarrow B, C \rightarrow cC, D \rightarrow D] \)

\( P_3 : [A \rightarrow A, B \rightarrow bB, C \rightarrow C, D \rightarrow dD] \)

\( P_4 : [A \rightarrow a, B \rightarrow b, C \rightarrow c, D \rightarrow d] \)
Some sample derivations are:

\[
S \Rightarrow ABCD \Rightarrow aABcCD \Rightarrow aabccdd
\]
\[
S \Rightarrow ABCD \Rightarrow aABcCD \Rightarrow aAbBcCdD \Rightarrow aabbbccdd
\]

We can see that the application of matrix \( P_2 \) produces an equal number of \( a \)'s and \( c \)'s, application of \( P_3 \) produces an equal number of \( b \)'s and \( d \)'s. \( P_4 \) terminates the derivation. Clearly

\[
L(G) = \left\{ a^n b^m c^n d^m \mid n, m \geq 1 \right\}.
\]

The rules in each matrix are context free, but the language generated is context-sensitive and not context-free.
Example 2

Let $G = (N, T, P, S)$ be a matrix grammar with

$N = \{S, A, B, C\}$

$T = \{a, b\}$

$P = \{P_1, P_2, P_3, P_4, P_5\}$, where

$P_1 : [S \rightarrow ABC]$

$P_2 : [A \rightarrow aA, B \rightarrow aB, C \rightarrow aC]$

$P_3 : [A \rightarrow bA, B \rightarrow bB, C \rightarrow bC]$

$P_4 : [A \rightarrow a, B \rightarrow a, C \rightarrow a]$

$P_5 : [A \rightarrow b, B \rightarrow b, C \rightarrow b]$
Some sample derivations are:

\[ S \Rightarrow ABC \Rightarrow aAaBaC \Rightarrow abAabBabC \Rightarrow abaabaaba \]

\[ S \Rightarrow ABC \Rightarrow bAbBbC \Rightarrow baAbaBbaC \Rightarrow babbabbab \]

clearly

\[ L(G) = \left\{ www \mid w \in \{a, b\}^+ \right\}. \]
Programmed Grammar

A Programmed Grammar is a 4-tuple \( G = (N, T, P, S) \) where \( N \), \( T \) and \( S \) are as in any Chomsky grammar. Let \( r \) be a collection of re-writing rules over \( N \cup T \), \( \text{lab}(R) \) being the labels of \( R \). \( \sigma \) and \( \varphi \) are mappings from \( \text{lab}(R) \) to \( 2^{\text{lab}(R)} \)

\[
P = \left\{ (r, \sigma(r), \varphi(r)) \mid r \in R \right\}
\]

Here, \( G \) is said to be type \( i \), or \( \varepsilon - \text{free} \) if the rules in \( R \) are all type \( i \), where \( i = 0,1,2,3 \) or \( \varepsilon - \text{free} \), respectively.
Definition 3

For any $x, y$ over $(N \cup T)^*$, we define derivation as below:

(i) $(u, r_1) \Rightarrow (v, r_2)$ if and only if $u = u_1 xu_2, v = u_1 yu_2$ for $u_1, u_2$ are over $N \cup T$, and $(r_1 : x \rightarrow y, \sigma(r_1), \varphi(r_1)) \in P$ and $r_2 \in \sigma(r_1)$ and

(ii) $(u, r_1) \Rightarrow_{ac} (v, r_2)$ if and only if $(u, r_1) \Rightarrow (v, r_2)$ holds, or else $u = v$ if $r_1 : (x \rightarrow y, \sigma(r_1), \varphi(r_1))$ is not applicable to $u$, i.e., $x$ is not a sub word of $u$ and $r_2 \in \varphi(r_1)$. Thus, $\Rightarrow_{ac}$ only depends on $\varphi$.

Here, $\sigma(r)$ is called the success field as the rule with label $r$ is used in the derivation step. $\varphi(r)$ is called the failure field as the rule with label $r$ cannot be applied and we move on to a rule with label in $\varphi(r)$. 
are the reflexive and transitive closures of $\Rightarrow_{ac}$ and $\Rightarrow_{ac}$, respectively.

The language generated is defined as follows:

$$L(G, \sigma) = \left\{ w \mid w \in T^*, (S_1, r_1) \Rightarrow^* (w, r_2) \text{ for some } r_1, r_2 \in \text{lab}(P) \right\}$$

$$L(G, \sigma, \varphi) = \left\{ w \mid w \in T^*, (S_1, r_1) \Rightarrow_{ac}^* (w, r_2) \text{ for some } r_1, r_2 \in \text{lab}(P) \right\}$$

Let $P(P_{ac})$ denote the family of programmed languages without (with) appearance checking of type 2 without $\varepsilon$-rules.

Let $P^\lambda(P_{ac})$ denote the family of programmed languages without (with) appearance checking of type 2 with $\varepsilon$-rules.
Example 3

Let $G = (N, T, P, S)$ be a programmed grammar with

$N = \{S, A, B, C, D\}$

$T = \{a, b, c, d\}$

$P$: 

<table>
<thead>
<tr>
<th></th>
<th>$r$</th>
<th>$\sigma(r)$</th>
<th>$\varphi(r)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$S \longrightarrow$ ABCD</td>
<td>2, 3, 6</td>
<td>$\phi$</td>
</tr>
<tr>
<td>2</td>
<td>$A \longrightarrow$ aA</td>
<td>4</td>
<td>$\phi$</td>
</tr>
<tr>
<td>3</td>
<td>$B \longrightarrow$ bB</td>
<td>5</td>
<td>$\phi$</td>
</tr>
<tr>
<td>4</td>
<td>$C \longrightarrow$ cC</td>
<td>2, 3, 6</td>
<td>$\phi$</td>
</tr>
<tr>
<td></td>
<td>( r )</td>
<td>( \sigma(r) )</td>
<td>( \varphi(r) )</td>
</tr>
<tr>
<td>---</td>
<td>----------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>5.</td>
<td>D ( \rightarrow ) dD</td>
<td>2,3,6</td>
<td>( \phi )</td>
</tr>
<tr>
<td>6.</td>
<td>A ( \rightarrow ) a</td>
<td>7</td>
<td>( \phi )</td>
</tr>
<tr>
<td>7.</td>
<td>B ( \rightarrow ) b</td>
<td>8</td>
<td>( \phi )</td>
</tr>
<tr>
<td>8.</td>
<td>C ( \rightarrow ) c</td>
<td>9</td>
<td>( \phi )</td>
</tr>
<tr>
<td>9.</td>
<td>D ( \rightarrow ) d</td>
<td>( \phi )</td>
<td>( \phi )</td>
</tr>
</tbody>
</table>
Let $lab(F) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Some sample derivations are

$$
S \Rightarrow ABCD \Rightarrow aBCD \Rightarrow abCD \Rightarrow abcD \Rightarrow abcd
$$

$$
S \Rightarrow ABCD \Rightarrow aABCD \Rightarrow aABcCD \Rightarrow aaBcCD
\Rightarrow aabcCD \Rightarrow aabccD \Rightarrow aabccd
$$

$$
L(G) = \left\{ a^n b^m c^n d^m \mid n, m \geq 1 \right\}
$$
Example 4

Let \( G = (N, T, P, S) \) be a programmed grammar with

\[
N = \{S, A, B, C\}
\]

\[
T = \{a, b\}
\]

\( P: \)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>( S \rightarrow ABC )</td>
<td>2,5,8,11</td>
</tr>
<tr>
<td></td>
<td>( A \rightarrow aA )</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( B \rightarrow aB )</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>( C \rightarrow aC )</td>
<td>2,5,8,11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>5.</td>
<td>A → bA</td>
<td>6</td>
</tr>
<tr>
<td>6.</td>
<td>B → bB</td>
<td>7</td>
</tr>
<tr>
<td>7.</td>
<td>C → cB</td>
<td>2,5,8,11</td>
</tr>
<tr>
<td>8.</td>
<td>A → a</td>
<td>9</td>
</tr>
<tr>
<td>9.</td>
<td>B → a</td>
<td>10</td>
</tr>
<tr>
<td>10.</td>
<td>C → a</td>
<td>φ</td>
</tr>
<tr>
<td>11.</td>
<td>A → b</td>
<td>12</td>
</tr>
<tr>
<td>12.</td>
<td>B → b</td>
<td>13</td>
</tr>
<tr>
<td>13.</td>
<td>C → b</td>
<td>φ</td>
</tr>
</tbody>
</table>

\[ L(G) = \left\{ w w w | w \in \{a,b\}^+ \right\}. \]
Random Context grammar

A Random context grammar has two sets of nonterminals $X$, $Y$ where the set $X$ is called the permitting context and $Y$ is called the forbidding context of a rule $x \rightarrow y$. 
Definition 4

\[ G = (N, T, P, S) \] is a random context grammar where \( N, T \) and \( S \) are as in any Chomsky grammar, where

\[ p = \left\{ (x \rightarrow y, X, Y) \mid x \rightarrow y \text{ is a rule over } N \cup T, X, Y \text{ are subsets of } N \right\} \]

We say \( u \Rightarrow v \) if and only if \( u = u' xu", v = u' yu" \) for \( u', u" \) over \( N \cup T \) such that all symbols \( X \) appear in and appears in \( u', u" \) and no symbol of \( Y \) appears in \( u', u" \). \( \Rightarrow^* \) is the reflexive transitive closure of \( \Rightarrow \).

\[ L(G) = \left\{ w : S \Rightarrow^* w, w \in T^* \right\}. \]

As before, \( L \) is of type \( i \), whenever \( G \) with rules \( x \rightarrow y \) in \( P \) are of type \( i \), \( i=0,1,2,3 \), respectively.
Example 5

Consider the random context grammar \( G = (N, T, P, S) \) where

\[
N = \{S, A, B, C\}
\]

\[
T = \{a\}
\]

\[
\begin{align*}
P &= \left\{ (S \rightarrow AA, \phi, \{B, D\}), (A \rightarrow B, \phi, \{S, D\}), \\
&\quad (B \rightarrow S, \phi, \{A, D\}), (A \rightarrow D, \phi, \{S, B\}), \\
&\quad (D \rightarrow a, \phi, \{S, A, B\}) \right\}.
\end{align*}
\]

Some sample derivations are

\[
S \Rightarrow AA \Rightarrow DA \Rightarrow DD \Rightarrow aD \Rightarrow aa
\]

\[
S \Rightarrow AA \Rightarrow BA \Rightarrow BB \Rightarrow SB \Rightarrow SS
\]

\[
\Rightarrow AAS \Rightarrow AAAA \Rightarrow a^4
\]

\[
L(G) = \left\{ a^{2^n} \mid n \geq 1 \right\}.
\]
Time varying Grammar

Given a grammar $G$, one can think of applying a set of rules only for a particular period. That is, the entire set of $P$ is not available at any step of a derivation. Only a subset of $P$ is available at any time ‘$t$’ or at any $i$-th step of a derivation.

**Definition 5**

A time-varying grammar of type $i$, $0 \leq i \leq 3$, is an ordered pair $(G, \phi)$ where $G = (N, T, P, S)$ is a type $i$ grammar, and $\phi$ is a mapping of the set of natural numbers into the set of subsets of $P$. $(u, i) \Rightarrow (v, j)$ holds if and only if:

1. $j = i + 1$ and

2. There are words $u_1, u_2, x, y$ over $N \cup T$ such that $u = u_1 xu_2$, $v = u_1 yu_2$ and $x \rightarrow y$ is a rule over $N \cup T$ in $\phi(i)$. 
* be the reflexive, transitive closure of and

\[ L(G, \phi) = \{ w \mid (S, 1) \Rightarrow (w, j) \} \quad \text{for some } j \in N, w \in T^* \]

A language \( L \) is time varying of type \( i \) if and only if for some time varying grammar \( (G, \phi) \) is of type \( i \) with \( L = L(G, \phi) \).
Definition 6

Let \((G, \phi)\) be a time varying grammar. Let \(F\) be a subset of the set of productions \(P\). A relation \(\Rightarrow_{ac}\) on the set of pairs \((u, j)\), where \(u\) is a word over \(N \cup T\) and \(j\) is a natural number which is defined as follows:

\[
\left( u, j_1 \right) \Rightarrow_{ac} \left( v, j_2 \right)
\]

holds, if

\[
\left( u, j_1 \right) \Rightarrow \left( v, j_2 \right)
\]

holds, or else,

\[
j_2 = j_1 + 1, u = v,
\]

and for no production

\[
x \rightarrow y \quad \text{in} \quad F \cap \phi(j_1)
\]

\(x\) is a subword of \(u\).
\( \rightarrow^{ac} \) is the reflexive, transitive closure of \( \rightarrow^{ac} \). Then, the language generated by \( (G, \phi) \) with appearance checking for productions in \( F \) is defined as:

\[
L_{ac}(G, \phi, F) = \left\{ w \mid w \in T^* \mid (S, 1) \rightarrow^{ac}(w, j) \text{ for some } j \right\}
\]

The family of languages of this form without appearance checking when the rules are context free (context-free and \( \varepsilon \text{-free} \)) and \( \phi \) is a periodic function are denoted as \( \tau^\lambda \) and \( \tau \), respectively. With appearance checking, they are denoted as \( \tau_{ac}^\lambda \) and \( \tau_{ac} \), respectively.
Example 6

Let \((G, \phi)\) be a periodically time varying grammar with

\[
G = (N, T, P, S) \quad \text{where}
\]

\[
N = \{S, X_1, Y_1, Z_1, X_2, Y_2, Z_2\}
\]

\[
T = \{a, b\}
\]

\[
P = \phi(1) \cup \phi(2) \cup \phi(3) \cup \phi(4) \cup \phi(5) \cup \phi(6) \quad \text{where}
\]

\[
\phi(1) = \{S \rightarrow aX_1aY_1aZ_1, S \rightarrow bX_1bY_1bZ_1, X_1 \rightarrow X_1, Z_2 \rightarrow Z_2\}
\]

\[
\phi(2) = \{X_1 \rightarrow aX_1, X_1 \rightarrow bX_2, X_2 \rightarrow aX_1, X_2 \rightarrow bX_2, X_1 \rightarrow \epsilon, X_2 \rightarrow \epsilon\}
\]

\[
\phi(3) = \{Y_1 \rightarrow aY_1, Y_1 \rightarrow bY_2, Y_2 \rightarrow aY_1, Y_2 \rightarrow bY_2, Y_1 \rightarrow \epsilon, Y_2 \rightarrow \epsilon\}
\]

\[
\phi(4) = \{Z_1 \rightarrow aZ_1, Y_1 \rightarrow bZ_2, Z_2 \rightarrow aZ_1, Z_2 \rightarrow bZ_2, Z_1 \rightarrow \epsilon, Z_2 \rightarrow \epsilon\}
\]
\[
\phi(5) = \{ X_2 \to X_2, Y_1 \to Y_1 \}
\]
\[
\phi(6) = \{ Y_2 \to Y_2, Z_1 \to Z_1 \}
\]

Some sample derivations are:

\[
(S,1) \Rightarrow (aX_1 aY_1 aZ_1,2) \Rightarrow (aaY_1 aZ_1,3) \Rightarrow (aaaZ_1,4) \Rightarrow (aaa,5)
\]
\[
(S,1) \Rightarrow (bX_1 bY_1 bZ_1,2) \Rightarrow (baX_1 bY_1 bZ_1,3) \Rightarrow (baX_1 baY_2 bZ_1,4)
\]
\[
\Rightarrow (baX_1 baY_1 baZ_1,5) \Rightarrow (baX_1 baY_1 baZ_1,6)
\]
\[
\Rightarrow (baX_1 baY_1 baZ_1,7) \Rightarrow (baX_1 baY_1 baZ_1,8)
\]
\[
\Rightarrow (babY_1 baZ_1,9) \Rightarrow (bababaZ_1,10)
\]
\[
\Rightarrow (bababa,11)
\]

\[
L(G,\phi) = \{ wwww \mid w \in \{a,b\}^+ \}
\]
Example 7

Let $(G, \phi)$ be a periodically time varying grammar with

\[ G = (N, T, P, S) \]

\[ N = \{ A, B, C, D, S, A_1, A_2, B_1, B_2, C_1, C_2, D_1, D_2 \} \]

\[ T = \{ a, b, c, d \} \]

\[ P : \bigcup_{i=1}^{8} \phi(i), \text{ where} \]

\[ \phi(1) = \{ S \rightarrow aA bB cC dD, D_1 \rightarrow D, A_2 \rightarrow A \} \]

\[ \phi(2) = \{ A \rightarrow aA_1, A_1 \rightarrow A_2, A \rightarrow \varepsilon \} \]
\[ \phi(3) = \{ B \rightarrow B_1, B \rightarrow bB_2, B \rightarrow \varepsilon \} \]
\[ \phi(4) = \{ C \rightarrow cC_1, C \rightarrow C_2, C \rightarrow \varepsilon \} \]
\[ \phi(5) = \{ D \rightarrow D_1, D \rightarrow dD_2, D \rightarrow \varepsilon \} \]
\[ \phi(6) = \{ A_1 \rightarrow A, B_2 \rightarrow B \} \]
\[ \phi(7) = \{ B_1 \rightarrow B, C_2 \rightarrow C \} \]
\[ \phi(8) = \{ C_1 \rightarrow C, D_2 \rightarrow D \} \]

\[ L(G, \phi) = \left\{ a^n b^m c^n d^m \mid n, m \geq 1 \right\} . \]
Regular Control Grammars

Let $G$ be a grammar with production set $P$ and $\text{lab}(P)$ be the labels of productions of $P$. To each derivation $D$, according to $G$, there corresponds a string over $\text{lab}(P)$ (the so-called control string). Let $C$ be a language over $\text{lab}(P)$. We define a language $L$ generated by a grammar $G$ such that every string of $L$ has a derivation $D$ with a control string from $C$. Such a language is said to be a controlled language.

Definition 7

Let $G = (N, T, P, S)$ be a grammar. Let $\text{lab}(P)$ be the set of labels of productions in $P$. Let $F$ be a subset of $P$. Let $D$ be a derivation of $G$ and $K$ be a word over $\text{lab}(P)$. $K$ is a control word of $D$, if and only if the following conditions are satisfied:
1. For some string $u,v,u_1,u_2,x,y$ over $N \cup T$, $D : u \Rightarrow v$ and $K=f$
   where $u = u_1 xu_2$ , $v = u_1 y u_2$ and $x \rightarrow y$ has a label $f$.

2. For some $u$, $x$, $y$, $D$ is a derivation of a word ‘u’ only and $K = \varepsilon$ or else
   $K = f$, where $x \rightarrow y$ has a label $f \in F$ and $x$ is not a sub word of $u$.

3. For some $u,v,w,K_1,K_2$, $D$ is a derivation $u \Rightarrow v \Rightarrow w$, where
   $K = K_1 K_2$ and $u \Rightarrow v$ uses $K_1$ as control string and $v \Rightarrow w$ uses $K_2$
   as control string.

Let $C$ be a language over the alphabet lab(P). The language generated by $G$
with control language $C$ with appearance checking rules $F$ is defined by :

$$L_{ac} \left( G, C, F \right) = \left\{ w \in T^* \mid D : S \Rightarrow^* w, D has a control word K of C \right\}$$
If $F = \emptyset$ the language generated is without appearance checking and denoted by $L(G,C)$.

Whenever $C$ is regular and $G$ is of type $i$, where $i = 0, 1, 2, 3$, then $G$ is said to be a regular control grammar of type $i$.

Let $\mathcal{L}(i, j, k)$ denote a family of type $i$ languages with type $j$ control with $k=0, 1$. $k=0$ denotes without appearance checking; $k=1$ denotes with appearance checking.
Example 8

Let \( G = (N, T, P, S) \) be a regular control grammar where

\[ N = \{ A, B, C, D, S \} \]
\[ T = \{ a, b, c, d \} \]
\[ P: \]
1. \( S \rightarrow ABC \)
2. \( A \rightarrow aA \)
3. \( B \rightarrow bB \)
4. \( C \rightarrow cC \)
5. \( D \rightarrow dD \)
6. \(A \rightarrow a\)
7. \(B \rightarrow b\)
8. \(C \rightarrow c\)
9. \(D \rightarrow d\)

Then, \(\text{lab}(P) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}\)

Let, \(K = \{1\}(24)^* (35)^* 6789\). Clearly, \(K\) is regular. Then

\(L(G,K) = \{a^n b^m c^n d^m \mid n, m \geq 1\}\)

Some sample derivations are:

for \(u = 124356789 \in K\),
\[
S \Rightarrow ABCD \Rightarrow aABCD \Rightarrow aABcCD \Rightarrow aAbBcCD
\]
\[
\Rightarrow aAbBcCdD \Rightarrow aabBcCdD \Rightarrow aabbcCcdD
\]
\[
\Rightarrow aabbcccdD \Rightarrow aabbcCcdD
\]

If \( u = 124246789 \in K \)

\[
S \Rightarrow ABCD \Rightarrow aABCD \Rightarrow aABcCD \Rightarrow aaAAbBcCD
\]
\[
\Rightarrow aaAAbBccCD \Rightarrow aaaaBccCD \Rightarrow aabbcCcCD
\]
\[
\Rightarrow aaabBcccdD \Rightarrow aaabbcCcdD
\]
Example 9

Let \( G = (N, T, P, S) \) be a grammar with

\( N = \{S, A, B, C\} \)

\( T = \{a, b\} \)

\( P: \)

1. \( S \rightarrow ABC \)
2. \( A \rightarrow aA \)
3. \( B \rightarrow aB \)
4. \( C \rightarrow aC \)
5. \( A \rightarrow bA \)
6. \( B \rightarrow bB \)
7. $c \rightarrow bC$
8. $A \rightarrow a$
9. $B \rightarrow a$
10. $C \rightarrow a$
11. $A \rightarrow b$
12. $B \rightarrow b$
13. $C \rightarrow b$

and $\text{lab}(P) = \{1, 2, \ldots, 13\}$

$K = 1(234 + 567)^* \left(89(10) + (11)(12)(13)\right)$

be a regular control on $G$.

$L(G, K) = \left\{www | w \in \{a, b\}^+ \right\}$
Indian Parallel Grammars

In the definition of matrix, programmed, time-varying, regular control, and random context grammars, only one rule is applied at any step of derivation. In this section, we consider parallel application of rules in a context-free grammars (CFG).

Definition 8

An Indian parallel grammar is a 4-tuple $G = (N, T, P, S)$ where the components are as defined for a CFG. We say that $x \Rightarrow y$ holds in G for strings $x, y$ over $N \cup T$, if

$$x = x_1 A x_2 A \ldots A x_n A x_{n+1}, \quad A \in N, \quad x_i \in (N \cup T) - \{ A \}^*$$

for $1 \leq i \leq n + 1$

$$y = x_1 w x_2 w \ldots w x_n w x_{n+1}, \quad A \rightarrow w \in P.$$
i.e., if a sentential form $x$ has an occurrences of the nonterminal $A$, and if $A \rightarrow w$ is to be used it is applied to all $A$’s in $x$ simultaneously. $\Rightarrow^*$ is the reflexive, transitive closure of $\Rightarrow$

$$L(G) = \left\{ w \mid w \in T^*, S \Rightarrow^* w \right\}$$
Example 10

We consider the Indian parallel grammar:

\[ G = \left( \{ S \}, \{ a \}, \{ S \rightarrow SS, S \rightarrow a \}, S \right). \]

Some sample derivations are

\[ S \Rightarrow a \]
\[ S \Rightarrow SS \Rightarrow aa, \]
\[ S \Rightarrow SS \Rightarrow SSSS \Rightarrow aaaaa \quad \text{and} \]
\[ L(G) = \left\{ a^{2^n} / n \geq 0 \right\}. \]

It is clear from this example that some non-context free languages can be generated by Indian parallel grammars.

The other way round, the question is: can all context free languages (CFL) be generated by Indian parallel grammars? Since the first attempt to solve this was made in (Siromoney and Krithivasan. 1974), this type of grammar is called an Indian parallel grammar.