## Grammar Systems and Distributed Automata

With the need to solve different problems within a short time in an efficient manner, parallel and distributed computing have become essential. Study of such computations in the abstract sense , from the formal-language theory point of view, started with the development of grammar systems . In classical formal-language theory, a language is generated by a single grammar or accepted by a single automation. They model a single processor or we can say the devices are centralized. Though multi tape Turing machines (TMs) try to introduce parallelism in a small way , the finite control of the machine is only one.

The introduction of distributed computing useful in analyzing computation in computer networks , distributed databases etc., has led to the notions such as distributed parallelism, concurrency , and communication. The theory of grammar systems and the distributed automata are formal models for distributed computing, where these notions could be formally defined and analyzed.

## CD Grammar Systems

### Definition

A CD grammar system of degree n ≥ 1, is a construct.:

$$GS = \left(N, T, S, P_1, \ldots\ldots, P_n\right)$$

Where N and T are disjoint alphabets ( non terminals and terminals) ;

$S \in N$ is the start symbol and $P_1, \ldots\ldots. P_n$ are the finite sets of rewriting rules over $N \cup T$ . $P_1, \ldots\ldots. P_n$ are called components of the system.

Another way of specifying a CD grammar system is :

$$GS = \left(N, T, S, G_1, \ldots\ldots, G_n\right)$$

where $G_i = \left(N, T, P_i, S\right), 1 \le i \le n.$

**Definition**

Let $GS = (N, T, S, P_1, \ldots\ldots P_n)$ be a CD grammar system. We now define different protocols of co-operation.

1. Normal mode $(*\ \text{mod}e) : \overset{*}{\underset{P_i}{\Rightarrow}}$ is defined by , $x \overset{*}{\underset{P_i}{\Rightarrow}} y$ without any restriction.

   The student works on the blackboard as long as he wants.

2. Terminating mode $(t\ \text{mod}e)$: for each $i \in \{1, \ldots\ldots n\}$, the terminating derivation by the ith component , denoted by $\overset{t}{\underset{P_i}{\Rightarrow}}$ , is defined by $x \overset{t}{\underset{P_i}{\Rightarrow}} y$ *if* and only *if* $x \overset{*}{\underset{P_i}{\Rightarrow}} y$ and there is no $z \in (N \cup T)^*$ with $y \underset{P_i}{\Rightarrow} z$.

3 . = k mode : For each $i \in \{1,......,n\}$ the k-steps derivation by the ith component, denoted by $\overset{=k}{\underset{P_i}{\Longrightarrow}}$, is defined by $x \overset{=k}{\underset{P_i}{\Longrightarrow}} y$ if and only if there are $x_1,.......x_{k+1} \in (N \bigcup T)^*$ such that $x = x_i$ , $y = x_{k+1}$ and for each j , $1 \leq j \leq k$

$$x_j \underset{P_i}{\Longrightarrow} x_{j+1}.$$

4. $\leq$ k mode : For each component $P_i$, the $\leq$ k – steps derivation by the ith component denoted by $\overset{\leq k}{\underset{P_i}{\Longrightarrow}}$, is defined by :

$$x \overset{\leq k}{\underset{P_i}{\Longrightarrow}} y \text{ if and only if } x \overset{=k'}{\underset{P_i}{\Longrightarrow}} y \text{ for some } k' \leq k.$$

5. $\geq$ k mode : The $\geq$ k steps of derivation by the ith component , denoted as $\overset{\geq k}{\underset{P_i}{\Rightarrow}}$, is defined by

$$x \overset{\geq k}{\underset{P_i}{\Rightarrow}} y \text{ if and only if } x \overset{=k'}{\underset{P_i}{\Rightarrow}} y \text{ for some } k' \geq k.$$

Let $D = \{*, t\} \cup \{\leq k, \geq k, = k \mid k \geq 1\}$.

**Definition**

The language generated by a CD grammar system

$GS = \left( N, T, S, P_1, \ldots\ldots, P_n \right)$ in derivation mode $f \in D$ is :

$$L_f\left(GS\right) = \left\{ \begin{array}{l} W \in T^* \mid S \underset{P_{i_1}}{\overset{f}{\Rightarrow}} \alpha_1 \underset{P_{i_2}}{\overset{f}{\Rightarrow}} \alpha_2 \ldots\ldots \underset{P_{i_m}}{\overset{f}{\Rightarrow}} \alpha_m = w, m \geq 1, \\ \qquad\qquad\qquad\qquad\qquad 1 \leq i_j \leq n, 1 \leq j \leq m \end{array} \right\}$$

**Example**

1. Consider the following CD grammar system :

$$GS_1 = \left(\left\{S, X, X', Y, Y'\right\}, \left\{a, b, c\right\}, S, P_1, P_2\right),$$

$$P_1 = \left\{S \to S, S \to XY, X' \to X, Y' \to Y\right\}$$

$$P_2 = \left\{X \to aX', Y \to bY'c, X \to a, Y \to bc\right\}$$

If $f = *$ mode , the first component derives $S \Rightarrow XY$ the second component derives from $Y, bY'c$, it can switch to first component or derive $aX'$ from X.

In the first component $X'$ can be changed to $X \; or \; Y'$ can be changed to Y or both . The derivation proceeds similarly.

It is not difficult to see that the language generated is

$$\left\{a^m b^n c^n \setminus m, n \geq 1\right\}.$$

The same will be true for

$$t \text{ mod} e, = 1 \text{ mod} e, \geq 1 \text{ mod} e, \leq k \text{ mod} e \; for \, k \geq 1.$$

But, if we consider = 2 mode , each component should execute two steps . In the first component $S \Rightarrow S \Rightarrow XY$ . In the second component , $XY \Rightarrow aX^{'}Y \Rightarrow aX^{'}bY^{'}c$.

Then control goes back to component one where $X^{'}$ and $Y^{'}$ are changed to X and Y in two steps. The derivation proceeds in the similar manner.

It is easy to see that the language generated by $GS_1$ in the = 2 mode is $\{a^n b^n c^n \mid n \geq 1\}$. A similar argument holds for $\geq 2$ – mode also and the language generated is the same .

At most , two steps of derivation can be done in each component . Hence , in the case of = k or ≥ k mode where k ≥ 3 , the language generated is the empty set.

2. $GS_2 = \left( \{S, A\}, \{a\}, S, P_1, P_2, P_3 \right)$

$$P_1 = \{S \rightarrow AA\}$$

$$P_2 = \{A \rightarrow S\}$$

$$P_3 = \{A \rightarrow a\}$$

In the * mode $\{a^n \mid n \geq 2\}$ is generated as the control can switch from component to component at any time.

A similar results holds for $\geq 1, \leq k\left(k \geq 1\right)$ modes. For $= 1, = k$, $\geq k\left(k \geq 2\right)$, the language generate is empty as $S \rightarrow AA$ can be used only once in $P_1$ and $A \rightarrow a$ can be used once in $P_3$.

In the ***t*** mode in $P_1$ ,$S \Rightarrow AA$ and if the control goes to $P_3$ from AA , $aa$ is derived . If the control goes to $P_2$ from AA , SS is derived . Now the control has to go to $P_1$ to proceed with the derivation

$SS \Rightarrow AAAA$ , and if the control goes to $P_2$ ,$S^4$ is derived ; if it goes to $P_3$ ,$a^4$ is derived . It is easy to see that the language generated in $t$ mode is

$$\left\{a^{2^n} \mid n \geq 1\right\}.$$

3. $GS_3 = \left(\left\{S, X_1, X_2\right\}, \left\{a, b\right\}, S, P_1, P_2, P_3\right),$

where

$$P_1 = \left\{S \to S, S \to X_1 X_1, X_2 \to X_1\right\}$$
$$P_2 = \left\{X_1 \to aX_2, X_1 \to a\right\}$$
$$P_3 = \left\{X_1 \to bX_2, X_1 \to b\right\}$$

In * mode , $= 1, \geq 1$ mode, $\leq k$ mode $(k \geq 2)$, $t$ mode the language generated will be $\left\{w \mid w \in \{a,b\}^*, |w| \geq 2\right\}$. In $= 2$ mode , each component has to execute two steps , so the language generated will be $\left\{ww \mid w \in \{a,b\}^+\right\}$.

A similar argument holds for $\geq 2$ steps . For $= k$ or $\geq k$ modes $(k \geq 3)$, the language generated is empty , as each component can use at most two steps before transferring the control.

We state some results about the generative power without giving proof. The proofs are fairly simple , and can be tried as exercise . It can be easily seen that for CD grammars systems working in any of the modes defined having regular , linear , context – sensitive , or type 0 components , respectively , the generative power does not change ; i.e., they generate the families of regular , linear, context – sensitive , or recursively enumerable languages , respectively .

But by the example given , we find that CD grammar systems with context – free components can generate context – sensitive languages. Let $CD_n(f)$ denote the family of languages generated by CD grammar systems with $\varepsilon - free$ context free components , the number of components being at most n . When the number of components is not limited , the family is denoted by $CD_\infty(f)$ if $\varepsilon - rules$ are allowed the corresponding families are denoted by

$CD_n^\varepsilon(f)$ $and$ $CD_\infty^\varepsilon(f)$, respectively .