SIMULATED LANGUAGE MODELS

P.Vijaya Vani[1]

Abstract: ***In this paper new language models called simulated language models are introduced. For developing these language models language paths, language connectivity are defined and equivalence class of languages is established. Further the language equiv-component of Languages in Recursive languages $Ł_R$ is established and proved that the language equi-component of Languages is the largest language connected subset of $Ł_R$ containing L. Also it is noted that a continuous image of language connected class is language connected.***

In addition Ordinal Recursive Language is defined by building a countable class of languages called ordinal class. Further it is proved that for any countable class $\mathcal{S}$, if $\Sigma^* \cup B \notin \mathcal{S}$ then Sup $\mathcal{S} \neq \Sigma^* \cup B$. Indentifying the order topology on $Ł_R$ by using base intervals of the type $(L_1, \Sigma^* \cup B)$, (L_1, L_2), $[1, L_1)$, $(L_2, \Sigma^* \cup B]$ for various $L_1, L_2 \in Ł_R$, the class of all ordinals with this order topology on it is called the ordinal class of recursive languages. Also it is proved that the uncountable ordinal recursive language $\Sigma^* \cup B$ has no countable neighborhood base.Further it is shown that ordinal class of recursive languages is compact and Hausdorff. It is also proved that every subspace class of the ordinal class of recursive languages $Ł_R$ is T_3 and Tychnoff. Further it is identified that the subspace class of ordinal class of recursive languages is countably compact and sequentially compact.

In order to deduce the simulated language model , Patterns, generalized patterns, constituents of a pattern, limit of the constituent, left context, right context, left terminal, right terminal are defined and finally for $\gamma \notin \Sigma$ the left or right terminal, a simulated language Model SLM denoted S of a Turing Machine is a finite collection of triplets (a, d, b) called essentials, where $a \in (\varepsilon \cup \gamma)\Sigma^*$, $b \in \Sigma^*(\gamma \cup \varepsilon)$ and $d \in (\Sigma \cup \{\lambda\})^* - \{\lambda\}$. The string λ is called the prototype of the essentials (a, d, b) and the strings a and b are called the tolerable left or right contexts. Further language of a simulated language model is recognized. In this process it is shown that there exists recursive languages that cannot be defined by a right linear simulated language models where in a simulated language model right linear means if all its essentials are of the form (a, bI, γ) for $a \in (\gamma \cup \varepsilon)\Sigma^*$, $b \in \Sigma^+$.

Finally it is proved that for each $h \geq 0$, recursive language L_h, there is a simulated language $L(S_h)$ in each level of the star height class structure.

Keywords: ordinal recursive language, ordinal class, simulated language models, star height class structure.

1. INTRODUCTION

Definition 1.1 Let $Ł_R$ be the class of recursive languages. If $f : [B, \Gamma^*] \rightarrow Ł_R$ is a continuous map such that f(B)= B and $f(\Gamma^*) = \Sigma^* \cup B$, we say that f is a language

ISBN - 978-93-81583-56-2

path joining B to $\Sigma^* \cup B$. Here B is called the initial point and $\Sigma^* \cup B$ the final point of path f.

Definition 1.2 The class of recursive languages $Ł_R$ is said to be language connected if for any two recursive languages in $Ł_R$ there exists a language joining them.

Definition 1.3 If $Ł_R$ is language connected then any two recursive languages in $Ł_R$ are connected. We write $L_i \approx L_j$ for $L_i , L_j \in Ł_R$ if there is a language in $Ł_R$ joining L_i and L_j.

Theorem 1.4 If $Ł_R$ is the class of recursive languages then $\approx$ is an equivalence relation on $Ł_R$.

Proof: For any $L_1 \in Ł_R$ by the definition of "language connectivity"

Clearly $L_1 \approx L_1$ Thus ' $\approx$ ' is reflexive.
Suppose $L_1 \approx L_2$

that implies there is a language path joining L_1 and L_2

$\Rightarrow$ there exist $f : [B , \Gamma^*] \rightarrow Ł_R$ a continuous map such that $f(B) = L_1$ and $f(\Gamma^*) = L_2$, we say that f is a language path joining L_1 to L_2.

$\Rightarrow$ f is a language path joining L_2 to L_1.

$\Rightarrow L_2 \approx L_1$. Thus ' $\approx$ ' is symmetric.

Suppose $L_1 \approx L_2$ and $L_2 \approx L_3$ that implies there is a language path joining L_1 and L_2 and a language path joining L_2 and L_3

$\Rightarrow$ there exist $f : [B , \Gamma_1^*] \rightarrow Ł_R$ a continuous map such that $f(B) = L_1$ and $f(\Gamma^*) = L_2$, we say that f is a language path joining L_1 to L_2 and there exist $g : [\Gamma_1^* , \Gamma^*] \rightarrow Ł_R$ a continuous map such that $f(\Gamma_1^*) = L_2$ and $f(\Gamma^*) = L_3$, we say that g is a language path joining L_2 to L_3

$\Rightarrow$ f is a language path joining L_1 to L_2 and g is a language path joining L_2 to L_3

$\Rightarrow$ gof is a language path joining L_1 to L_3

$\Rightarrow L_1 \approx L_3$. Thus ' $\approx$ ' is transitive.

Therefore ($Ł_R, \approx$) is an equivalence relation.

Definition 1.5 For $L \in Ł_R$, let $\xi_L = \{ L_* \in Ł_R / L_* \approx L \}$ be the equivalence class of L in $Ł_R$ under $\approx$. ξ_L is called the language equiv-component of L in $Ł_R$.

Theorem 1.6 Let $L \in Ł_R$ be an arbitrary recursive language in $Ł_R$ and ξ_L be its language equiv-component. Then ξ_L is the largest language connected subset of $Ł_R$ containing L.

Proof: If $L_1, L_2 \in Ł_R$, there is a language L_i in $Ł_R$ joining L_1 to L and a language L_j in $Ł_R$ joining L to L_2. Then the language L_iL_j in $Ł_R$ join L_1 to L_2 . First we show that L_iL_j is a language in ξ_L. By 3.20 clearly $\exists$ w in $\Sigma^* \cup B$ such that $qw \vdash^* \alpha_1 p \alpha_2$ for q the initial state, p the final state and $\alpha_1\alpha_2$ in Γ^* . Thus there is a language, for completeness sake say L_iL_j in ξ_L joining L_1 to L_2 . Hence ξ_L is language connected. Therefore ξ_L is the largest language connected subset of $Ł_R$ containing L.

Let ζ_L be a language connected set containing L and let $L^* \in \zeta_L$. Then there is a language L_λ in ζ_L joining L to L*.

Let $\rho : \zeta_L \to Ł_R$ be the continuous map $\rho(L^*)=L^*$ for $L^* \in \zeta_L$. Then $\rho(L^*)L_\lambda$ is a language in ξ_L joining L to L*, so $L^* \in \xi_L$. Thus $\zeta_L \subset \xi_L$.

Remark 1.7 A continuous image of language connected class is language connected.

2. THE ORDINAL CLASS OF RECURSIVE LANGUAGES

Definition 2.1 Let $Ł_R$ be the class of recursive languages. If

(1) $Ł_R$ is an uncountable linearly ordered set under a linear order $\leq$. For $L_1 < L_2$ if $L_1 \leq L_2$ and $L_1 \neq L_2$.

(2) $Ł_R$ is well ordered under its given order $\leq$. Thus every non-empty class $\subseteq Ł_R$ has the least element in it.

(3) $Ł_R$ has the greatest element which is denoted by $\Sigma^* \cup B$.

Thus $L \leq \Sigma^* \cup B$ for every $L \in Ł_R$

(4) For every $L \in Ł_R$ if $L \neq \Sigma^* \cup B$, then $\{ L_i \in Ł_R / L_i \leq L\}$ is a countable class.

then the class $Ł_R$ is called the ordinal class and every $L \in Ł_R$ is called an ordinal recursive language.

Definition 2.2 If for $L_1, L_2 \in Ł_R$, $L_1 < L_2$ we say that L_1 is a predecessor of L_2. If $L_1 < L_2$ and there is no $L \in Ł_R$ such that $L_1 < L < L_2$, we say L_1 is an immediate predecessor of L_2 or that L_2 is an immediate successor of L_1.

Remark 2.3 For $L \in Ł_R$, if $L \neq \Sigma^* \cup B$, the class of predecessors of L is countable.

Note 2.4 Since $Ł_R$ is well ordered, every non-empty class of recursive languages in $Ł_R$ has the predecessor B as the least element.

ISBN - 978-93-81583-56-2

Definition 2.5 For any non-empty class of recursive languages $\mathcal{S} \subset Ł_R$, let $Ł_{\mathcal{S}} = \{ L \in Ł_R / L_i \leq L \ \ \forall \ \ L_i \in \mathcal{S} \}$. Since $\Sigma^* \cup B \in Ł$ $Ł_{\mathcal{S}}$ is non-empty and has the least element which would then be Sup $\mathcal{S}$. Thus every nonempty class has both the least element as well as supremum.

Note 2.6 In any non-empty class of recursive languages $\mathcal{S} \subset Ł_R$, the least element is always in $\mathcal{S}$ and Sup $\mathcal{S}$ may or may not be in $\mathcal{S}$.

Proposition 2.7 For any countable class $\mathcal{S}$, if $\Sigma^* \cup B \notin \mathcal{S}$ then Sup $\mathcal{S} \neq \Sigma^* \cup B$.

Proof: Let $Ł_R^{\dagger} = \{ L \in Ł_R / L \leq L_i$ for some $L_i \in \mathcal{S} \}$

$= \cup \{ L \in Ł_R / L \leq L_i \}$ $L_i \in \mathcal{S}$

Thus $Ł_R^{\dagger}$ is a countable class and if $L^* =$ Sup $Ł_R^{\dagger}$, $Ł_R^{\dagger}$ is the set of predecessors of L^* and by definition of ordinal class of Recursive languages $L^* \neq \Sigma^* \cup B$.

But Sup $\mathcal{S} \leq L^* < \Sigma^* \cup B$. This proves that Sup $\mathcal{S} \neq \Sigma^* \cup B$.

Definition 2.8 The least element of the whole class of recursive languages $Ł_R$ is called the first ordinal recursive language and is denoted by 1. The set $Ł_R - \{1\}$ has the least element; it is called the second ordinal recursive language and is denoted by 2. Thus $1 < 2$ and 1 is the immediate predecessor of 2 while 2 is the successor of 1 and so on. After defining the n^{th} ordinal recursive language and denoting it by n, the least element of $Ł_R - \{ 1, 2 , 3,\ldots,n\}$ is the $(n+1)^{th}$ ordinal recursive language and is denoted by $n + 1$ and $1 < 2 < 3 < \ldots\ldots < n < n + 1 < \ldots\ldots\ldots$ Then the set $\mathbb{N}$ of positive integers identified with first ordinals preserving their natural order. Every such ordinal n has a successor and except for 1 every other ordinal recursive language n has immediate predecessor. Each $n \in \mathbb{N}$ is called a finite ordinal recursive language.

Definition 2.9 The set $Ł_R - \mathbb{N}$ is non-empty and has the least element which is greater than every finite ordinal n. It is denoted by Δ. It is the first non finite ordinal recursive language.

In generalization, every $n \in \mathbb{N}$ is a predecessor of Δ but Δ has no immediate predecessor. Also, every ordinal recursive language except $\Sigma^* \cup B$ has the successor and every ordinal except 1 has a predecessor but may not have an immediate predecessor.

Definition 2.10 An ordinal recursive language number (other than 1) which has no immediate predecessor is called a limit ordinal recursive language. Thus Δ is the first ordinal recursive language limit in $Ł_R$.

Remark 2.11 The process of obtaining successors may be repeated beyond Δ , the successor of Δ will be denoted by $\Delta + 1$, its successor by $\Delta + 2$ and so on. Thus we visualize a copy of $\mathbb{N}$ with its linear order beyond Δ, another copy of $\mathbb{N}$ beyond $\Delta + \Delta$ and so on.

Definition 2.12 The class of predecessors of each of the ordinal recursive languages appearing in this sequence is countable i.e none of these ordinal recursive languages equals $\Sigma^* \cup B$. Thus $\Sigma^* \cup B$ is called the first uncountable ordinal recursive language. $\Sigma^* \cup B$ is a limit ordinal recursive language.

Note 2.13 The class $Ł_R$ of ordinal recursive languages is thus an uncountable, linearly ordered, well ordered class with the order defined.

Notation 2.14 For $L_1, L_2 \in Ł_R$ with $L_1 < L_2$ we write

$(L_1, L_2) = \{ L \in Ł_R / L_1 < L < L_2 \}$

$[L_1, L_2] = \{ L \in Ł_R / L_1 \leq L \leq L_2 \}$

$(L_1, \Sigma^* \cup B] = \{ L \in Ł_R / L_1 < L \}$

$[1, \Sigma^* \cup B] = Ł_R$ where 1 is the first ordinal recursive language.

$[1, \Sigma^* \cup B) = \{ L \in Ł_R / L < \Sigma^* \cup B \} = Ł_R - \{\Sigma^* \cup B \}$

$[1, \Delta) = \mathbb{N}$

$[1, \Delta] = \{ L \in Ł_R / L \leq \Delta \}$

Definition 2.15 The order topology on $Ł_R$ is obtained by using as a base intervals of the type $(L_1, \Sigma^* \cup B)$, (L_1, L_2), $[1, L_1)$, $(L_2, \Sigma^* \cup B]$ for various $L_1, L_2 \in Ł_R$ .The class of all ordinals with the order topology on it is called the ordinal class of recursive languages.

Remark 2.16 Clearly by 1. $Ł_R$ is a Hausdorff class of recursive languages.

Theorem 2.17 The uncountable ordinal recursive language $\Sigma^* \cup B$ has no countable neighborhood base.

Proof: Suppose $\{LB_j\}$ is any countable class of basic neighborhoods of $\Sigma^* \cup B$. If it is a base, since $Ł_R$ is Hausdorff, $\cap LB_j = \{ \Sigma^* \cup B\}$. On the other hand, each $LB_j = (L_j , \Sigma^* \cup B)$ for some $L_j \in Ł_R$ with $L_j < \Sigma^* \cup B$. For each j , the class of the predecessors of L_j is countable, since $L_j \neq \Sigma^* \cup B$, $\text{Sup} \{ L_j / j = 1,2,\ldots\ldots\} = L_S < \Sigma^* \cup B$ and so $(L_S , \Sigma^* \cup B]$ is uncountable.

For each j, $(L_S , \Sigma^* \cup B] \subset (L_j , \Sigma^* \cup B]$, so $(L_S , \Sigma^* \cup B] \subset \cap LB_j$, an uncountable class. This is a contradiction. This shows that $Ł_R$ is not first countable at $\Sigma^* \cup B$. Since $Ł_R$ is not first countable, it is not second countable either.

Theorem 2.18 The class $Ł_R$ of ordinal recursive languages is compact and Hausdorff.

Proof: Let $\mathbb{O}$ be a basic open cover for $Ł_R$ .Then $\Sigma^* \cup B \in U$ for some $U \in \mathbb{O}$. Let O_1 be the least ordinal such that $\Sigma^* \cup B \in (O_1, \Sigma^* \cup B] = U_1$ for some $U_1 \in \mathbb{O}$. Clearly such an O_1 exists and $O_1 < \Sigma^* \cup B$. If the first ordinal $1 \neq O_1$, let O_2 be the least ordinal $\ni O_1 \in (O_2, O_1] = U_2$ for some $U_2 \in \mathbb{O}$. The sequence $\{ O_1, O_2,\ldots\ldots\}$

ISBN - 978-93-81583-56-2

thus obtained satisfies $O_1 > O_2 > \ldots\ldots$ and has the least element say O_n. Thus $O_n \in Ł_R$ is such that $(O_n, O_{n-1}] \subset U_n$ for some $U_n \in \mathbb{O}$. The finite class $\{ U_1, U_2, \ldots\ldots\ldots U_n \}$ covers all of $Ł_R$ except possibly the ordinal 1. Let $U_0 \in \mathbb{O}$ be such that $1 \in U_0$. Then $\{ U_0, U_1, \ldots\ldots\ldots U_n \}$ is a finite sub cover for $Ł_R$ from $\mathbb{O}$. Thus $Ł_R$ is Hausdorff.

Corollary 2.19 $Ł_R$ is T_3 and T_4. Every subspace class of the ordinal class of recursive languages $Ł_R$ is T_3 and Tychnoff.

Remark 2.20 For each $L \in Ł_R$, $[1, L]$ is a closed ordinal class of recursive languages $Ł_R$ and therefore compact. In particular $[1, \Delta]$ is compact but $[1, \Delta) = \mathbb{N}$ is not compact. In fact , the subspace class $[1, \Delta)$ is countably infinite and discrete. Thus $[1, \Delta)$ is not compact but $[1, \Delta]$ the closure of $[1, \Delta)$ is compact.

Also the subspace $Ł_R{}^0 = [1, \Sigma^* \cup B)$ is an open set of $Ł_R$ with closure of $Ł_R{}^0 = Ł_R$. Thus $Ł_R{}^0$ is not compact.

Theorem 2.21 The subspace class $Ł_R{}^0$ of $Ł_R$ is countably compact and sequentially compact.

Proof: If a countable open cover $\mathbb{O} = \{ U_1, U_2, \ldots\ldots\ldots U_n \}$ of $Ł_R{}^0$ has no finite sub cover, for each n, there is ordinal $O_n < \Sigma^* \cup B$ such that $O_n \notin U_1 \cup U_2 \cup \ldots\ldots\ldots \cup U_n$. Then $\text{Sup} \{ O_1, O_2, \ldots\ldots\ldots O_n \} = O_S < \Sigma^* \cup B$ and the construction of O_S would imply that for the compact class $[1, O_S]$ there is no finite sub cover from the open cover $\mathbb{O}$. This contradiction shows that $Ł_R{}^0$ must be countably compact . Since $Ł_R{}^0$ is first countable, sequential compactness of $Ł_R{}^0$ follows.

Remark 2.22 Since each basic open interval class is also closed it follows that the ordinal class of recursive languages and its subspace classes are 0-dimensional and hence totally disconnected.

Remark 2.23 The class $Ł_R = [1, \Sigma^* \cup B]$ and $[1, \Delta]$ are compact and Hausdorff. Therefore the product space $[1, \Sigma^* \cup B] \times [1, \Delta]$ denoted by T is called the Turing Tychonoff Plank.

Remark 2.24 Since the Tychonoff Plank T is compact and Hausdorff, it is also completely regular and normal.

3. SIMULATED LANGUAGE MODEL

Definition 3.1 Let $M = (Q, \sum, \Gamma, \delta, q_0 , B, F)$ be a Turing Machine and $\lambda \notin \Sigma$ be the placeholder. A Pattern T is a tree of M such that the transitional states of T are denoted by λ and the leaves of T are denoted by symbols from $\Sigma \cup \{ \varepsilon \}$. A generalized pattern T is a tree of M such that the transitional states of T are denoted by λ and the leaves of T are denoted by symbols from $\Sigma \cup I$, where $I = \{ \varepsilon, \lambda \}$.

Definition 3.2 The constituents C of a pattern are its sub trees of height one and leaves denoted from $\Sigma \cup \{ \varepsilon \} \cup \{ \lambda \}$.

Note 3.3 Let T be a pattern. For a transitional state q of T, let T_q and C_q be respectively the maximal sub tree and the constituent of T with starting state q .

Definition 3.4 Let T_1 be a tree obtained from the maximal sub tree T_q of T by deleting only the starting state q of T_q. Let a, b $\in \Sigma^*$ be two strings such that the limit of T_1 denoted $\ell(T_1) = a\lambda b$, where λ is the transitional state.

The limit of the constituent C_q is denoted by $\ell(C_q)$.

Notation 3.5 The left context of T_q and C_q of T are $L(T_q ,T) = L(C_q ,T) = a$ and the right context of T_q and C_q of T are $R(T_q , T) = R(C_q ,T) = b$.

Definition 3.6 Let $\gamma \notin \Sigma$ be the left or right terminal. A simulated language Model SLM denoted S of a Turing Machine is a finite collection of triplets (a, d, b) called essentials, where $a \in (\varepsilon \cup \gamma) \Sigma^*$, $b \in \Sigma^*(\gamma \cup \varepsilon)$ and $d \in (\Sigma \cup \{\lambda\})^* - \{ \lambda \}$. The string λ is called the prototype of the essentials (a, d, b) and the strings a and b are called the tolerable left or right contexts.

Definition 3.7 Let S be a simulated language model. A constituent C_q of a pattern T is connected by an essential (a, d, b) iff

(i). a is a suffix of $\gamma L(C_q, T)$.

(ii). $d = \ell (C_q)$

(iii). b is a prefix of $R(C_q, T) \gamma$.

A pattern T is said to be normal in a simulated language model S iff each constituent C_q of T is connected by an essential of S.

Definition 3.8 The language defined by a simulated language model denoted L(S) is the set $\{ s \in \Sigma^* / s = \ell(T)$ for some $T \in P(S)\}$ where P(S) is the set of all patterns normal for S.

Notation 3.9 Suppose c, e are two strings and A, B , D are finite sets of strings, the notation (A, cDe, B) denotes the set of essentials that is $\{(a, cde, b)/ a \in A, b \in B, d \in D \}$.

Definition 3.10 A simulated language model is right linear if all its essentials are of the form (a, bI, γ) for $a \in (\gamma \cup \varepsilon)\Sigma^*$, $b \in \Sigma^+$.

Theorem 3.11 There exists recursive languages that cannot be defined by a right linear simulated language models.

ISBN - 978-93-81583-56-2

Proof: Consider L = {x/x ∈ { a, b }*, $|x|_a = 1$} where $|x|_a$ gives the number of times the symbol a in the string x occurred. Suppose that here is a right linear simulated language model L(S) for L of degree d > 0 and length l > 0. Consider a string $sb^{d+2l}ab^{d+2l}$ corresponding to a pattern T. Since the simulated language model is right linear, there are two constituents C , $C^{\dagger}$ of T such that $\ell(C) \in b^{+}\lambda$, $\ell(C^{\dagger}) \in b^{*}ab^{*}\lambda$ and $L(C, T) = L(C^{\dagger}, T) \in b^{*}b^{d}$, with $R(C, T) = R(C^{\dagger}, T^{l}) = \gamma$. Hence, the pattern T^{l} obtained by replacing $C^{\dagger}$ in T with C is normal for L(S), but $(T^{l}) \in b^{+}$ and hence is not in L.

Definition 3.12 For every $x \in \Sigma \cup \gamma^{*}$, prefix $_d$ (x) denotes the prefix of length d of x if $|x| \geq d$, otherwise it denotes x; suffix $_d$ (x) is the suffix of x of length d if $|x| \geq d$, otherwise x. The two operations are extended to every pattern T as prefix $_d$ (T) = prefix $_d$ ((T)), suffix $_d$ (T) = suffix $_d$ ((T)).

Definition 3.13 Let T be a pattern and T_* be one of its maximal sub trees then $L_d(T_*, T)$ = suffix $_d$ $(L(T_*, T)$ i.e., the left context of length d of T_* in T. Similarly $R_d(T_*, T)$ = prefix $_d$ $(R(T_*, T)$ i.e., the right context of length d of T_* in T.

Remark 3.14 Let L(S) be a simulated language model of degree d ≥ 0 and let T and T^{l} be two normal trees of L(S), not necessarily distinct. Suppose that there exist two maximal sub trees T_* of T and T^{l}_* of T^{l} such that such that $L(T_*, T) = L_d(T^{l}_*, T^{l})$, $R(T_*, T) = R_d(T^{l}_*, T^{l})$, prefix $_d$ (T_*) = prefix $_d$ (T^{1}_*), suffix $_d$ (T_*) = suffix $_d$ (T^{1}_*).

Then the pattern T* obtained from T by replacing the sub tree T_* with the sub tree T^{l}_* is a normal tree of L(S).

Definition 3.15 A n-expression string $exp_n(x)$ is a string x in a language with a set of at least n ≥ 0 positions of x.

Given a recursive language L , an integer n ≥ 0 and a n-
expresed string $exp_n(x)$, x ∈ L and $|x| \geq n$, a partition of x is
a quintuple $(x_1 , x_2 , x_3 , x_4 , x_5)$ such that
$x = x_1 x_2 x_3 x_4 x_5$
x_2 and x_4 together correspond to atleast one-expressed position in $exp_n(x)$.
$x_2 x_3 x_4$ correspond to at most n-expressed position in $exp_n(x)$.
for every i ≥ 0 $x_1 x^{i}_2 x_3 x^{i}_4 x_5 \in L$.

Lemma 3.16 For a recursive language L there exists an integer n > 0 such that for all n-expressed strings $exp_n(x)$, with x ∈ L and $|x| > n$, there exists a partition.

Definition 3.17 Two partitions $(x_1 , x_2 , x_3 , x_4 , x_5)$ and $(y_1 , y_2 , y_3 , y_4 , y_5)$ are d-closer if the following conditions hold for some i, j ≥ 0.

1) $suffix_d(x_1) = suffix_d(y_1)$
2) $prefix_d(x^{i}_2 x_3 x^{i}_4) = prefix_d(y^{j}_2 y_3 y^{j}_4)$

3) $suffix_d(x^i_2\, x_3\, x^i_4) = suffix_d(y^j_2\, y_3\, y^j_4)$
4) $prefix_d(x_5) = prefix_d(y_5)$

Remark 3.18 In any simulated language model $L(S) \subseteq \Sigma^*$ of degree $d \geq 0$ there exists an integer $n > 0$ such that for any given two n-expressed strings $exp_n(x)$, $exp_n(y)$ $x, y \in L$ and $|x| > n$, $|y| > n$, if all pairs of partition for $exp_n(x)$, $exp_n(y)$ are d-closer, then for every pair of partition $(x_1, x_2, x_3, x_4, x_5)$ and $(y_1, y_2, y_3, y_4, y_5)$ of $exp_n(x)$, $exp_n(y)$ respectively $x_1\, y^k_2\, y_3\, y^k_4\, x_5 \in L \subseteq L(S)$ for every $k \geq 0$.

Definition 3.19 For each $h \geq 0$, recursive language L_i on a binary alphabet defined recursively as $\gamma_{i+1} = ({}_a2^h\gamma_{hb}2^h\gamma_h)^*$, $h \geq 0 \gamma_0 = \varepsilon$ is said to have star height h.

Theorem 3.20 For each $h \geq 0$, recursive language L_h, there is a simulated language $L(S_h)$ in each level of the star height class structure.

Proof: Claim: L_h is defined by the simulated language $L(S_h)$.

Suppose $L(S_h)$ is defined by the following essentials $\{(\gamma, {}_{Ia}2^{h-1}Q_{h-1I}, \varepsilon),(\gamma, \varepsilon, \gamma)\} \cup \bigcup_{1\leq i<h} \{(\gamma_a2^{h-1}\, {}_a2^{h-2} \ldots{}_a2^{h-i}{}_{,I\,a}2^{h-i-1}\, Q_{h-i-1I}, \varepsilon)\} \cup \bigcup_{1\leq i<h\,;\, h-i\leq k<h} \{({}_{ba}2^k\, {}_a2^{k-1} \ldots{}_a2^{h-i}{}_{,I\,a}2^{h-i-1}$ $Q_{h-i-1I}, \varepsilon) \cup \bigcup_{1\leq i<h\,;\, 2^{h-i}\leq j<2^{h-i+1}} \{({}_{ab}{}^j{}_{,I\,a}2^{h-i-1}\, Q_{h-i-1I}, \varepsilon)$

where $Q_h = I_b2^h$ if $h > 0$ and $Q_0 = b$.

$L(S_h(\gamma)) = \{(\gamma_a2^{h-1}\, {}_a2^{h-2} \ldots{}_a2^{h-i}{}_{,I\,a}2^{h-i-1}\, Q_{h-i-1I}, \varepsilon)/\ 1 \leq i < h\}$
$L(S_h(a)) = \{({}_{ab}{}^j{}_{,I\,a}2^{h-i-1}\, Q_{h-i-1I}, \varepsilon)/\ 1 \leq i < h\,;\ 2^{h-i} \leq j < 2^{h-i+1}\}$
$L(S_h(b)) = \{({}_{ba}2^k\, {}_a2^{k-1} \ldots{}_a2^{h-i}{}_{,I\,a}2^{h-i-1}\, Q_{h-i-1I}, \varepsilon)/1 \leq i < h\,;\ \ h\text{-}i \leq k < h\}$
Let $\Delta_h(p, q, \varepsilon) = ({}_{pa}2^h, q, \varepsilon)$ $\forall$ $(p, q, \varepsilon) \in L(S_h(\gamma)) \cup L(S_h(b))$ and finally let

$\Delta_h(\gamma, {}_{Ia}2^{h-1}Q_{h-1I}, \varepsilon) = \{(\gamma_a2^h, {}_{Ia}2^{h-1}Q_{h-1I}, \varepsilon),({}_{ba}2^h, {}_{Ia}2^{h-1}Q_{h-1I}, \varepsilon)\} \cup \{({}_{pb}2^h, {}_{Ia}2^{h-1}Q_{h-1I}, \varepsilon)/(p, {}_{Ia}2^{h-2}Q_{h-2I}, \varepsilon) \in L(S_h(a))\}$ Then the simulated language's $L(S_h)$ can be recursively defined as

$L(S_1) = \{(\gamma, IabI, \varepsilon),(\gamma, \varepsilon, \gamma)\}$

$L(S_{h+1}) = \{(\gamma, {}_{Ia}2^{h-1}Q_{h-1I}, \varepsilon),(\gamma, \varepsilon, \gamma)\} \cup L(S_h(a)) \cup L(S_h(b)) \cup \Delta_h(\gamma, {}_{Ia}2^{h-1}Q_{h-1I}, \varepsilon) \cup (p, q, \varepsilon) \in L(S_h(\gamma)) \cup L(S_h(b))\ \Delta_h(p, q, \varepsilon)$ Now we prove by induction on h that $L_h = L(S_h)$

For $h = 1$, Clearly $L_1 = (ab)^*$ is a simulated language.

Suppose that $L_{h-1} = L(S_{h-1})$. Now we show that $L_h \subseteq L(S_h)$.

For all $x \in L_h$, $x \in L(S_h)$ by induction on $|x|$. Since the empty string and the shortest non-empty string in L_h , ${}_a2^{h-1}\, {}_b2^{h-1}$ both belong to L_h and to $L(S_h)$. Let $x = \prod_{1\leq i<n} ({}_a2^{h-1}\, s_{ib}2^{h-1}\, t_i)$ where $\prod$ denotes concatenation $s_i, t_i \in L_{h-1}$, $1 \leq i < n$ and $s_i\, t_i \neq \varepsilon$ if $n = 1$

ISBN - 978-93-81583-56-2

Suppose n = 1. If $t_1 \neq \varepsilon$ then $x_0 = {}_a2^{h-1}\, s_{1b}2^{h-1}$ is a string of L_h smaller than x, so there exist a normal tree $T(x_0)$ of $L(S_h)$ for x_0 and a generalized normal tree T_0 of $L(S_h)$, with $\ell(T_0) = x_0\lambda$.

Moreover $t_1 \in L_{h-1}$ and hence there is a normal tree $T(t_1)$ of $L(S_{h-1})$ for t_1. Numbering the states 0,1,2,……., let j be the leaf of T_0 denoted by λ. Extending $T(t_1)$ to the vertex j in T_0 is again a tree T with $\ell(T) = x$.

To prove that T is a normal tree of $L(S_h)$, we prove that each constituent of the maximal sub tree T_j, whose starting state is j, is connected by an essential of $L(S_h)$. Since t_j is a copy of $T(t_1)$, each constituent C of t_j has in $T(t_1)$ a copy connected by an essential $E_{h-l}(C) \in L(S_{h-1})$.

Moreover $L(C, T) = x_0 s$ with $L(C, T(t_1)) = \gamma s$ and either $s = \varepsilon$ or prefix ${}_a2^{h-2}(s) = {}_a2^{h-2}$. Hence if $E_{h-1}(C) \in L(S_{h-1}(a)) \cup L(S_{h-1}(b)) \subset L(S_h)$, C is connected by the same essential of $L(S_h)$ in T. Suppose $E_{h-1}(C) \in L(S_{h-1}(\lambda)) \cup \{(\gamma, {}_{Ia}2^{h-2}Q_{h-2I}, \varepsilon)\}$. If $E_{h-1}(C)$ is the essential $(\gamma_a2^{h-2}\, {}_a2^{h-3}\, {}_{.....a}2^{h-i-1}\, {}_{,I\,a}2^{h-i-2}\, Q_{h-i-2I}, \varepsilon)$ for some i, $1 \leq i < h-1$, then $L(C, T) = x_{0a}2^{h-2}\, {}_a2^{h-3}\, {}_{.....a}2^{h-i-1}$ and since suffix$_1(x_0) = b$, C is connected in T by the essential $({}_{ba}2^{h-2}\, {}_a2^{h-3}\, {}_{.....a}2^{h-i-1}\, {}_{,I\,a}2^{h-i-2}\, Q_{h-i-2I}, \varepsilon) \in L(S_h(b)) \subset L(S_h)$.

Otherwise if $E_{h-1}(C)$ is the essential $(\gamma, {}_{Ia}2^{h-2}Q_{h-2I}, \varepsilon)$ then $L(C, T) = x_0$ and since suffix ${}_2h\,(x_0) = {}_a{}^q\, {}_b2^h - q$, for some $q > 0$ such that $2^h - q \geq 2^{h-1}$, then C is connected by the essential $({}_{ab}2^h - q, {}_{Ia}2^{h-2}Q_{h-2I}, \varepsilon) \in \Delta_{h-1}(\gamma, {}_{Ia}2^{h-2}Q_{h-2I}, \varepsilon) \subset L(S_h)$. Hence each constituent C of T is connected by some essential of $L(S_h)$ and therefore T is a normal tree of $L(S_h)$.

If $t_1 = \varepsilon$, then $x_0 = {}_a2^{h-1}s_{1b}2^{h-1}$ with $s_1 \neq \varepsilon$, $s_1 \in L_{h-1}$.

Hence there is a normal tree $T(s_1)$ for $s_1 \in L(S_{h-1})$. Then let T_0 be the generalized pattern of $L(S_h)$, formed by the constituent connected by the initial essential $(\gamma, {}_a2^{h-1}\lambda\, {}_b2^{h-1}, \varepsilon)$. Joining $T(s_1)$ to the placeholder of T_0, we get a tree T with $\ell(T) = x$. Each constituent C of $T(s_1)$ is connected by an essential $E_{h-l}(C) \in L(S_{h-1})$. Again suppose $E_{h-l}(C) \in L(S_{h-1}(\gamma)) \cup \{(\gamma, {}_{Ia}2^{h-2}Q_{h-2I}, \varepsilon)\}$. If $E_{h-l}(C)$ is the essential $(\gamma_a2^{h-2}\, {}_a2^{h-3}\, {}_{.....a}2^{h-i-1}\, {}_{,I\,a}2^{h-i-2}\, Q_{h-i-2I}, \varepsilon)$ for some i, $1 \leq i < h-1$, then $L(C, T) = {}_a2^{h-1}\, {}_a2^{h-2}\, {}_a2^{h-3}\, {}_{.....a}2^{h-i-1}$ and C is connected by the essential $(\gamma_a2^{h-1}\, {}_a2^{h-2}\, {}_{.....a}2^{h-1-1}\, {}_{,I\,a}2^{h-i-2}\, Q_{h-i-2I}, \varepsilon) \in \Delta_{h-1}(S_{h-1}(\gamma)) \subset L(S_h)$.

Otherwise if $E_{h-1}(C)$ is the essential $(\gamma, {}_{Ia}2^{h-2}Q_{h-2I}, \varepsilon)$ then $L(C, T) = {}_a2^{h-1}$ and C is connected by the essential $(\gamma_a2^{h-1}, {}_{Ia}2^{h-2}Q_{h-2I}, \varepsilon) \in \Delta_{h-1}(\gamma, {}_{Ia}2^{h-2}Q_{h-2I}, \varepsilon) \subset L(S_h)$. Then in any case, each constituent C of T is connected by some essential of $L(S_h)$ and therefore T is again a normal tree of $L(S_h)$.

Consider the case $n \geq 1$ $x_1 = {}_a2^{h-1}s_{1b}2^{h-1}t_1$, $x_2 = \prod_{2 \leq i < n}({}_a2^{h-1}s_{ib}2^{h-1}t_i)$ are closer than x and belong to L_h; by induction, there are two valid trees $T(x_1)$ and $T(x_2)$ of $L(S_h)$ such that $\ell(T(x_1)) = x_1$, $\ell(T(x_2)) = x_2$. $T(x_2)$ has a unique constituent C connected by an essential of the form (γ, s, ε) with $s \in \{{}_a2^{h-2}\, {}_{Ib}2^{h-2}\, {}_I\}$ Replacing C with a constituent C^1 connected by the essential (γ , λs , ε) we get a generalized pattern T_2 whose maximal sub tree is λx_2. Let j be the unique leaf T_2 denoted by a placeholder. Joining $T(x_1)$ to the vertex j of T_2 given in the tree T with (T) = $x_1x_2 = x$.

Claim: T is a normal tree.

Let T_1 be the maximal sub tree of T of starting state/vertex j. T_2 is a copy of $T(x_1)$ so each constituent C joined to a vertex or state of T_1 has the same left context of its copy in $T(x_1)$; it is to be connected by some essential of $L(S_h)$. Now let C_1 be a constituent of T joined to a node n > j not belonging to the set of state/vertex of T_1 C_1 is a constituent of T_2 ; it is connected in T_2 by an essential E_h (C_l) of $L(S_h)$. Obviously $L(C_1, T) = x_1s$ where $L(C_1, T_2) = \gamma s$ and $prefix_2^{h-1}$ (s)= $_a2^{h-1}$. So if $E_h(C_l)$ $\in$ $L(S_h(a)) \cup L(S_h(b))$ then C_1 is connected by $E_h(C_l)$ also in T. Let $E_h(C_1) \in L(S_h(\gamma)) \cup \{(\gamma, {}_{Ia}2^{h-1}Q_{h-i-1I}, \varepsilon)\}$. If $E_h(C_1)$ is the essential $(\gamma_a 2^{h-1}{}_a 2^{h-2}{}_{.....a}2^{h-i}{}_{,I\,a}2^{h-i-1}\, Q_{h-i-1I}, \varepsilon)$ for some i, $1 \leq i \leq h$, then $L(C_1, T) = x_{1a}2^{h-1}{}_a2^{h-2}{}_a2^{h-3}{}_{.....a}2^{h-i}$ and since $suffix_1(x_1) = b$, C is connected in T by the essential $({}_{ba}2^{h-1}{}_a2^{h-2}{}_{.....a}2^{h-i}{}_{,\,I\,a}2^{h-i-1}\, Q_{h-i-1I}, \varepsilon)$ $\in L(S_h(b)) \subset L(S_h)$. Otherwise if $E_{h-1}(C_1)$ is the essential $(\gamma, {}_{Ia}2^{h-1}Q_{h-1I}, \varepsilon)$ then $L(C_1, T) = x_1$ and since suffix $_2h$ $(x_0) = {}_a{}^q{}_b2^h - q$, for some q > 0 such that $2^h - q \geq 2^{h-1}$, then C_{1} is connected in T by the essential $({}_{ab}2^h - q, {}_{Ia}2^{h-1}Q_{h-1I}, \varepsilon) \in L(S_h(a)) \subset L(S_h)$. Hence each constituent C_{1} of T is connected by some essential of $L(S_h)$ and therefore T is a normal tree of $L(S_h)$.

So $L_h \subset L(S_h)$ for all $h \geq 0$. On the other side we now prove $L(S_h) \subset L_h$. We prove this by induction on h. Let T be a normal tree of $L(S_h)$. If T is a normal tree of $L(S_h)$. If T has height 1, (T) = $_a2^{h-1}{}_b2^{h-1} \in L_h$. Suppose $(T^1) \in L_h$ for each normal tree T^1 of $L(S_h)$ of height j < n and let T be a normal tree of height n. The constituent C_0 of T with the same starting state/vertex 0 of T is connected by an essential (γ, $I_a2^{h-1}I$ $_b2^{h-1}$ I, ε). Joining all the vertices in T, the maximal sub tree T_i of T results by joining T to the vertex i. T_1 is a normal tree of $L(S_h)$ of height $j \leq n - 1$, hence $x_1 = \ell(T_1) \in L_h$, $T2+2^{h-1}$, $T\,3+2^h$ are normal trees of $L(S_{h-1})$ called T_2 and T_3.

Hence $x_2 = (T_2)$, $x_3 = (T_3) \in L_{h-1}$. So the limit of T is $x_{1a}2^{h-1}\, x_{2\,b}2^{h-1}\, x_3$ is obviously is in L_h. Therefore $L(S_h) \subset L_h$
Hence $L_h = L(S_h)$.

Therefore in each level of the tree height there is a Simulated Language Model.

4. REFERENCES

1. Adaricheva, K.V., and J.B.Nation. "Reflections on lower bounded lattices", *Algebra Universalis* 53(2005).307-330.
2. Beauquier.D. and Pin. J.E . "*Factors of words*". Volume 372 of Lecture notes in computer Science, pages 63-79, Berlin,1989. Spring Verlag.
3. Bernstein E,Vazrani, "*Quantum complexity theory*". SIAMJ comp,1997.
4. Campeanu.C. and Ho. Nu.W.H.,Culik.K. "State complexity of basic operations on finite languages". *In O. Boldt and H.Jurgensen, Proceedings of the 4th International workshop on implementing. Automata*, number 2214 in LNCS,

5. Campeanu.C. and Ho. W.H. "The maximum state complexity for finite language" *.Journal of Automata*, Languages and Combinatorics,9(2-3) 189-202,september2004.
6. Castro JL, Delgado M, Mantas C T. "*A new approach for the execution and adjustment of a fuzzy algorithm"* Fuzzy set system 2001.
7. Champarnadh.J.M. and Pin. J.E ."*A maximum problem on finite automata*". Discrete applied mathematics.
8. Crespi areghizzi.S., Pradella.M.,Pietro.P. "Associative definitions of programming languages". *Computer languages*,26:105-123,2000.
9. Eilenberg.S. "*Automata, Languages, and machines*", Vol. A. Acd Press,1974.
10. Feiner.l. *"Hierarchies of Boolean algebras"*.
11. Freese,R.,Jezek.J. amd Nation.J.B."Free Lattices", *Mathematical Surveys and Monographs*, vol. 42, Amer.Math.soc., Providence, RI, 1995.
12. Gile C,Omlin C,Thornber K K. *"Equivalence in knowledge representation: Automata, Recurrent neural network and dynamical fuzzy systems."*
13. Gratzer,G.,"*General Lattice Theory*", SecEdn,Birkhauser Verlag, Basel, 1998
14. Gunter Pilz."*Near-rings*"
15. Hanf.W. "Model-theoretic methods in the study of elementary logic", *Theory of models (Proc.1963 internet. Sympos., Berkeley)*, North-Holland,Amsteerdam,1965,pp.132-145.MR 35#1457.
16. Holcombe."*Near-rings associated with automata*", San B163-166.
17. Holzer.M. and M.Kutrib. *"State complexity of basic operations on nondeterministic finite automata"*
18. Hopcroft J.E,Ullman *"Introduction to Automata theory. Language and Computation."*
19. Hopcroft.J.E. and Ullman.J.D. "*Introduction to automata theory, Languages, and computation"*,addision-Weslwy,1979.
20. Joshi.A.K., Levy.L.S., and Yueh.K. *"Local constraints in the syntax and semantics of programming languages"*.
21. Lee E T,Zadeh L.A "A note on fuzzy languages" *inf Sci*, 1969.
22. Li.Y M,Pedrycz W.F. *"Fuzzy finite automata and fuzzy regular expressions with membership values in lattice -ordered monoids."*
23. Li.Y.M., Z.K.Shi, *"Finite Automata with values in lattice-monoid and their languages."*
24. Mordeson J N,Malik *"Fuzzy Automata and Languages: Theory and application."*
25. Nicaud.C "*Average state complexity of operations on unary automata*."
26. Omaratzki. M.,Kisman.D. "On the number of distinct languages accepted by finite automata with n states". *Journal of automata, languages*
27. Pedrycz W.Gomide "*An introduction to Fuzzy sets*": Cambridge mit press.
28. Repnitskii, V.B., "On finite lattices embedded in sub semi group lattices", *Semi group forum* 46(1993),388-397.
29. Salomaa. A., "*Theory of Automata*". Pergamum Press,Oxford,1969.
30. Schols.H.M.J.L. "The maximum number of states after projection", *Computing Science notes* No.87/06, Eindhoven University of technology, Dept. of. Math. And Comp. Sci., The Netherlands, April 1987.
31. Semenova. M.V., "On lattices embeddable in to suborder lattices". *Algebra and Logic* 44(2005),270-285.
32. Sivak, B., "Representation of finite lattices by orders on finite sets", *Math.*

****

----

[1]P.Vijaya Vani
Visiting Professor, Dept of IBS
Acharya Nagarajuna University , Guntur, Andhra Pradesh, India.