Considerations Regarding Formal Languages Generation Using Labelled Stratified Graphs

Tudor (Preda) Irina-Valentina

Abstract—The concept of labelled stratified graph was introduced in order to obtain the concept of knowledge base with output and various applications of this concept were presented ever since. This paper studies another application of this structure: generating formal languages by means of labelled stratified graphs. Various mechanisms to define and generate formal languages are known and we show that we can obtain different types of languages such as: regular languages or context sensitive language. We also give an example of context sensitive language, but not a context-free language (according to Chomsky hierarchy), that can be generated by labelled stratified graphs. The concepts introduced in this paper can initiate a possible research line concerning the generative power of the formal languages generated by labelled stratified graphs.

Keywords—context-free language, interpretation, labelled graph, labelled stratified graph, Peano algebra, regular language.

I. Introduction

The concept of *labelled stratified graph* was introduced in [10]. The application of this structure includes the domains:

- semantics of communication [10];
- image synthesis [10];
- reconstruction of a graphical image by extracting the semantics of a linguistic spatial description given in a natural language [7], [11];
- the modelling of the fusion action for two companies [12], [8];
- solving the problems which can be transposed in attribute graphs or colored graphs [12];
- knowledge bases with output and their use to the scheduling problems [4].

Various mechanisms for language generation are known: finite automata [5], regular expressions ([5], formal grammars [5], Lindenmayer systems [1]. Formal languages may be classified in the Chomsky hierarchy based on the expressive power of their generative grammar as well as the complexity of their recognizing automaton.

In this paper we show that we can generate formal languages by means of the labelled stratified graphs. Thus we obtain a new mechanism to generate formal languages, distinct of all known mechanisms used to generate such entities.

This paper is organized as follows: in Section II we recall the main concepts related to labelled stratified graphs; in Section III we show the manner in which we can generate formal languages by means of the labelled stratified graphs; the last section contains the conclusions and open problems.

I. Tudor(Preda), University of Pitesti, Department of Computer Science, email: irisxiphium@yahoo.com

This article was produced under the project: "Development of the doctoral schools by providing fellowships for young PhD students with frequency"-ID 52826

II. Labelled stratified graphs: short presentation

A. Definition of labelled stratified graphs

We consider a non empty set A. By a *binary partial operation* on A we understand a partial mapping f from $A \times A$ to A. This means that f is defined for the elements of some set $dom(f)$, where $dom(f) \subset A \times A$. We shall use the notation $f : dom(f) \longrightarrow A$. In the case when $dom(f) = A \times A$ we say that f is a *binary operation* on A.

We shall write $f \prec g$ if $f : dom(f) \longrightarrow A$ and $g : dom(g) \longrightarrow A$ are two functions such that $dom(f) \subseteq dom(g)$ and $f(x) = g(x)$ for all $x \in dom(f)$.

By a *partial σ-algebra* we understand a pair [2] $\mathcal{A}$=(A, σ_A), where A is the support set of $\mathcal{A}$ and σ_A is *a partial binary operation* on A. If $dom(\sigma_A) = A \times A$ then we say that $\mathcal{A}$ is a *σ-algebra*.

We consider a non-empty set S. If $\rho_1 \in 2^{S\times S}$ and $\rho_2 \in 2^{S\times S}$ then we define $\rho_1 \circ \rho_2$ as the set of all pairs $(x, y) \in S\times S$ for which there is $z \in S$ such that $(x, z) \in \rho_1$ and $(z, y) \in \rho_2$. We introduce the mapping

$$prod_S : dom(prod_S) \longrightarrow 2^{S\times S}$$

where $prod_S(\rho_1, \rho_2) = \rho_1 \circ \rho_2$ and $(\rho_1, \rho_2) \in dom(prod_S)$, if and only if and $\rho_1 \circ \rho_2 \neq \emptyset$.

By a *labelled graph* we understand a tuple
$G = (S, L_0, T_0, f_0)$ where:
S is the set of nodes,
L_0 is a finite set of labels,
$T_0 \subseteq 2^{S\times S}$ is a set of binary relations on S and
$f_0 : L_0 \longrightarrow T_0$ is a surjective mapping.

We denote by $R(prod_S)$ the set of all restrictions of the mapping $prod_S$:

$$R(prod_S) = \{u \mid u \prec prod_S\}$$

We observe that if u is an element of $R(prod_S)$ then the pair $(2^{S\times S}, u)$ is a partial algebra. This is a partial algebra used to obtain the structure named labelled stratified graph.

Take $u \in R(prod_S)$ and consider the closure $T = Cl_u(T_0)$ of T_0 in the algebra $(2^{S\times S}, u)$.

For each nonempty set M there is a Peano σ-algebra over M. Two Peano σ-algebras are isomorphic algebras and for this reason we shall use the following structure. We consider the set B given by

$$B = \bigcup_{n\geq 0} B_n \tag{1}$$

where

$$\begin{cases} B_0 = M \\ B_{n+1} = B_n \cup W_n \end{cases} \tag{2}$$

where

$$W_n = \{\sigma(x_1, x_2) \mid x_1, x_2 \in B_n\}$$

and $\sigma(x_1, x_2)$ is the word $\sigma x_1 x_2$ over the alphabet $\{\sigma\} \cup M$. The pair $PA(M) = (B, \sigma)$ is a Peano σ-algebra over M.

We consider some collection of subsets of B, denoted by $Initial(M)$. Namely, we say that $L \in Initial(M)$ if the following conditions are fulfilled:

- $M \subseteq L \subseteq B$
- if $\sigma(u, v) \in L$ then $u \in L$ and $v \in L$

Generally speaking, if $L \in Initial(M)$ then the pair (L, σ_L), where

- $dom(\sigma_L) = \{(x, y) \in L \times L \mid \sigma(x, y) \in L\}$
- $\sigma_L(x, y) = \sigma(x, y)$ for every $(x, y) \in dom(\sigma_L)$

is a partial σ-algebra.

We consider the Peano σ-algebra $PA(L_0) = (B, \sigma)$ over L_0, where B is given by (1) and (2) for $M = L_0$.

A **labelled stratified graph** $\mathcal{G}$ over G (shortly, **stratified graph** or LSG) is a tuple (G, L, T, u, f) where:

- $G = (S, L_0, T_0, f_0)$ is a labelled graph
- $L \in Initial(L_0)$
- $u \in R(prod_S)$ and $T = Cl_u(T_0)$
- $f : (L, \sigma_L) \longrightarrow (2^{S \times S}, u)$ is a morphism of partial algebras such that $f_0 \prec f$, $f(L) = T$ and if $(f(x), f(y)) \in dom(u)$ then $(x, y) \in dom(\sigma_L)$

We denote by $Strat(G)$ the set of all $LSGs$ over G. As we proved in [6] we have $Strat(G) \neq \emptyset$. Moreover, we proved in [12] that for every $u \in R(prod_S)$ there is just one stratified graph $\mathcal{G} = (G, L, T, u, f)$ over G and this structure is obtained applying the following steps [12]:

- Take a labelled graph $G = (S, L_0, T_0, f_0)$
- Take $u \in R(prod_S)$
- Compute $T = Cl_u(T_0)$
- Take $\{B_n\}_{n \geq 0}$ as in (2) for $M = L_0$
- Take $D_0 = L_0$ and define for every natural number $n \geq 0$ the entities (3), (4) and (5):

$$D_{n+1} = \{\sigma(p, q) \in B_{n+1} \setminus B_n \mid p, q \in dom(f_n), (f_n(p), f_n(q)) \in dom(u)\} \quad (3)$$

$$dom(f_{n+1}) = dom(f_n) \cup D_{n+1} \quad (4)$$

$$f_{n+1}(x) = \begin{cases} f_n(x) & \text{if } x \in dom(f_n) \\ u(f_n(p), f_n(q)) & \text{if } x = \sigma(p, q) \in D_{n+1} \end{cases} \quad (5)$$

- Define the mapping $f : dom(f) \longrightarrow T$ as follows:
  $dom(f) = \bigcup_{n \geq 0} dom(f_n) = \bigcup_{k \geq 0} D_k$
  $f(x) = f_k(x)$ if $x \in D_k$, $k \geq 0$
- Take $L = dom(f)$

B. Structured path and accepted structured path

We consider a path $d = ([x_1, ..., x_{n+1}], [a_1, ..., a_n])$ in a labelled graph $G = (S, L_0, T_0, f_0)$.

Consider the least set $STR(d)$ satisfying the following conditions:

- $([x_i, x_{i+1}], a_i) \in STR(d)$, $i \in \{1, ..., n\}$
- if $([x_i, ..., x_k], b_1) \in STR(d)$ and $([x_k, ..., x_r], b_2) \in STR(d)$, where $1 \leq i < k < r \leq n + 1$, then $([x_i, ..., x_r], [b_1, b_2]) \in STR(d)$

The maximal length elements of $STR(d)$, namely, the elements of the form $([x_1, ..., x_{n+1}], c) \in STR(d)$ are called **structured paths** over d.

Let d be a path. We define the mapping $h : STR_2(d) \longrightarrow B$ where B is defined in (1), as follows:

- $h(x) = x$ for $x \in L_0$
- $h([u, v]) = \sigma(h(u), h(v))$

The structured path $d_s \in STR(d)$ is named an **accepted structured path** over $\mathcal{G}$ if $d_s = ([x_1, ..., x_{n+1}], c)$ and $h(c) \in L$.

We denote by $ASP(\mathcal{G})$ the set of all accepted structured paths over $\mathcal{G}$. We denote by R a set of conditions imposed on the accepted structured paths. An element of $ASP(\mathcal{G})$ that satisfies R is named R-accepted structured path. We denote by $ASP_R(\mathcal{G})$ the set of all R-accepted structured paths.

For every accepted structured path $d = ([x_1, \ldots, x_{n+1}], \sigma(v_1, v_2)) \in ASP(\mathcal{G})$, where $n \geq 2$, there is one and only one $i \in \{2, \ldots, n\}$ such that $([x_1, \ldots, x_i], v_1) \in ASP(\mathcal{G})$ and $([x_i, \ldots, x_{n+1}], v_2) \in ASP(\mathcal{G})$ [10]. In other words, this property states that every accepted structured path over $\mathcal{G}$ can be broken into two accepted structured paths over $\mathcal{G}$. The number i stated in this property is named the *break index* for the path d_s and is denoted by $ind(d)$.

C. Interpretations of labelled stratified graphs

An **interpretation** for $\mathcal{G}$ is a tuple

$$\Sigma = (Ob, i, D, \mathcal{P})$$

where:

- Ob is a finite set of objects such that $Card(Ob) = Card(S)$
- $i : S \longrightarrow Ob$ is a bijective mapping
- $D = (Y, *)$ is a partial algebra, Y is called the domain of Σ and $*$ is a partial binary operation on Y
- $\mathcal{P} = \{Alg_a\}_{a \in L_0}$, where $Alg_a : Ob \times Ob \longrightarrow Y$

The **valuation mapping** generated by Σ is the mapping $val_\Sigma : ASP_R(\mathcal{G}) \longrightarrow Y$ defined inductively as follows:

$$\begin{cases} val_\Sigma([x, y], a) = Alg_a(i(x), i(y)) \\ val_\Sigma(x(1; n+1), \sigma(v_1, v_2)) = val_\Sigma(x(1; i), v_1) * \\ \qquad val_\Sigma(x(i; n+1), v_2) \end{cases}$$

where $i = ind([x_1, ..., x_{n+1}], \sigma(v_1, v_2))$ and $x(i; j) = [x_i, ..., x_j]$.

Consider a labelled stratified graph $\mathcal{G} = (G_0, L, T, u, f)$ over $G_0 = (S, L_0, T_0, f_0)$ and $\Sigma = (Ob, i, D, \mathcal{P})$ an interpretation for $\mathcal{G}$. A pair $(x, y) \in S \times S$ is called **interrogation**. For a given interrogation (x, y) we designate by $ASP_R(x, y)$ the set of all R-accepted structured paths from x to y in $\mathcal{G}$. The **answer mapping** is the mapping

$$Ans : S \times S \longrightarrow Y \cup \{no\}$$

defined as follows:

$$\begin{cases} Ans(x,y) = no \text{ if } ASP_R(x,y) = \emptyset \\ Ans(x,y) = \{val_\Sigma(d) \mid d \in ASP_R(x,y)\} \text{ if} \\ \qquad ASP_R(x,y) \neq \emptyset \end{cases}$$

III. Languages generated by a given labelled stratified graph

A formal language is a set of words, finite strings of letters, symbols or tokens. The set from which these letters are taken is the alphabet over which the language is defined.

In what follows we consider:

- An alphabet V.
- A stratified graph $\mathcal{G} = (G, L, T, u, f)$, where $G = (S, L_0, T_0, f_0)$ is a labelled graph.
- An interpretation $\Sigma = (V^*, i, D, \mathcal{P})$, where
  - V^* is the free monoid generated by V;
  - $i : S \longrightarrow V^*$
  - $D = (V^*, *)$ is a partial algebra, where $* : V^* \times V^* \longrightarrow V^*$ is a partial binary operation;
  - For each $a \in L_0$ we have $Alg_a : V^* \times V^* \longrightarrow V^*$ and $\mathcal{P} = \{Alg_a\}_{a \in L_0}$
- A subset $M \subseteq S \times S$
- The set R of restrictions to build the R-accepted structured paths.

Remark 3.1: Because $D = (V^*, *)$ is a partial algebra we can say that $val_\Sigma(x(1;i), v_1) * val_\Sigma(x(i;n+1), v_2) \in V^*$ if and only if $m_1 = val_\Sigma(x(1;i), v_1) \in V^*$, $m_2 = val_\Sigma(x(i;n+1), v_2) \in V^*$ and $(m_1, m_2) \in dom(*)$.

By definition, the language defined by the sets M and R is the following collection

$$\mathcal{L}(M,R) = \bigcup_{(x,y) \in M} Ans(x,y)$$

Remark 3.2: We observe that for each $(x,y) \in M$ the set $Ans(x,y) \subseteq V^*$ is a formal language over V, therefore $\mathcal{L}(M,R) \subseteq V^*$ is a formal language over V.

In order to exemplify the computations we consider the alphabet $V = \{d, e\}$ and the following labelled stratified graph $\mathcal{G} = (G, L, T, u, f)$, where G has the graphical representation from Fig. 1.

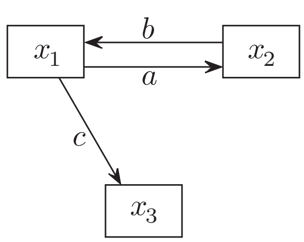

Figure 1. A labelled graph

In order to obtain the components of $\mathcal{G}$ we take $S = \{x_1, x_2, x_3\}$ and $L_0 = \{a, b, c\}$. We consider the binary relations:

$\rho_1 = \{(x_1, x_2)\}$
$\rho_2 = \{(x_2, x_1)\}$
$\rho_3 = \{(x_1, x_3)\}$.

We take $T_0 = \{\rho_1, \rho_2, \rho_3\}$ and we have to define the mapping u which defines uniquely the components of $\mathcal{G}$. We consider the mapping $u = prod_S$ defined in Table I, where:

$\rho_4 = \{(x_1, x_1)\}$
$\rho_5 = \{(x_2, x_2)\}$
$\rho_6 = \{(x_2, x_3)\}$.

u	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6
ρ_1		ρ_4			ρ_1	ρ_3
ρ_2	ρ_5		ρ_6	ρ_2		
ρ_3						
ρ_4	ρ_1		ρ_3	ρ_4		
ρ_5		ρ_2			ρ_5	ρ_6
ρ_6						

Table I
THE MAPPING u

We obtain $T = Cl_u(T_0) = \{\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6\}$. We shall suppose that $ASP_R(\mathcal{G}) = ASP(\mathcal{G})$. In other words, every accepted structured path is an R-accepted structured path.

Consider the interpretation $\Sigma = (V^*, i, D, \mathcal{P})$ for $\mathcal{G}$, where:
$i(x_1) = d$, $i(x_2) = e$, $i(x_3) = de$.
$\mathcal{P} = \{Alg_a, Alg_b, Alg_c\}$, where:
$Alg_a(x,y) = xy$
$Alg_b(x,y) = yx$
$Alg_c(x,y) = y$.

We define the operation $*$ in the following manner: if x, y are two words from V^*, then $x * y$ is the word xy.

Because the set L of this labelled stratified graph is infinite, the language defined by the labelled stratified graph using the interpretation Σ is the infinite set $\{(de)^n \mid n \geq 1\}$, where $(de)^n$ is de repeated n times.

Computing the valuation mapping for the accepted structured path $([x_1, x_2, x_1, x_3], \sigma(\sigma(a,b),c))$, we obtain:

$val_\Sigma([x_1, x_2, x_1, x_3], \sigma(\sigma(a,b),c)) =$
$val_\Sigma([x_1, x_2, x_1], \sigma(a,b)) * val_\Sigma([x_1, x_3], c) =$
$(val_\Sigma([x_1, x_2], a) * val_\Sigma([x_2, x_1], b)) * val_\Sigma([x_1, x_3], c) =$
$(Alg_a(i(x_1), i(x_2)) * Alg_b(i(x_2), i(x_1))) *$
$Alg_c(i(x_1), i(x_3)) = (Alg_a(d,e) * Alg_b(e,d)) * Alg_c(d, de) =$
$(de * de) * de = dede * de = dedede = (de)^3$.

If we consider the accepted structured path $([x_2, x_1, x_2, x_1, x_3], \sigma(\sigma(b,a), \sigma(b,c)))$, we obtain:

$val_\Sigma([x_2, x_1, x_2, x_1, x_3], \sigma(\sigma(b,a), \sigma(b,c))) =$
$(Alg_b(i(x_2), i(x_1)) * Alg_a(i(x_1), i(x_2))) *$
$(Alg_b(i(x_2), i(x_1)) * Alg_c(i(x_1), i(x_3))) =$
$(Alg_b(e,d) * Alg_a(d,e)) * (Alg_b(e,d) * Alg_c(d,de)) =$
$(de * de) * (de * de) = dede * dede = dededede = (de)^4$.

For this interpretation and the language generated by it we observe that the number of times the sequence "de" is repeated is given by the number of arcs covered by each accepted structured path computed.

Finally we obtain the answering mapping: $Ans(x_1,x_2) = Ans(x_2,x_1) = Ans(x_1,x_3) = \{(de)^{2k+1} \mid k \geq 0\}$; $Ans(x_1,x_1) = Ans(x_2,x_2) = Ans(x_2,x_3) = \{(de)^{2k} \mid k \geq 1\}$ and therefore we have

$$\mathcal{L}(M,R) = \{(de)^n \mid n \geq 1\}$$

for the case $M = \{(x_1,x_2),(x_2,x_1),(x_1,x_3),(x_1,x_1),(x_2,x_2),(x_2,x_3)\}$.

We can also see that $Ans(x_3,x_1) = Ans(x_3,x_2) = Ans(x_3,x_3) = \{no\}$ because $ASP(x_3,x_1) = ASP(x_3,x_2) = ASP(x_3,x_3) = \emptyset$ as no arc is leaving the node x_3.

If we consider the interpretation: $\Sigma_1 = (V^*, i, D_1, \mathcal{P}_1)$, where:

$i(x_1) = d$, $i(x_2) = dd$, $i(x_3) = e$,

$\mathcal{P}_1 = \mathcal{P}$

and the operation $*$ defined for the previous interpretation, when we compute the accepted structured path that we considered in the previous example we obtain:

$$\begin{aligned}
&val_{\Sigma_1}([x_2,x_1,x_2,x_1,x_3], \sigma(\sigma(b,a),\sigma(b,c))) = \\
&(Alg_b(i(x_2),i(x_1)) * Alg_a(i(x_1),i(x_2))) * \\
&(Alg_b(i(x_2),i(x_1)) * Alg_c(i(x_1),i(x_3))) = \\
&(Alg_b(dd,d) * Alg_a(d,dd)) * (Alg_b(dd,d) * Alg_c(d,e)) = \\
&(ddd*ddd)*(ddd*e) = dddddd*ddde = ddddddddde = d^9e.
\end{aligned}$$

The answering mapping is:

$Ans(x_1,x_2) = Ans(x_2,x_1) = \{d^{3(2k+1)} \mid k \geq 0\}$

$Ans(x_1,x_1) = Ans(x_2,x_2) = \{d^{3(2k)} \mid k \geq 1\}$

$Ans(x_1,x_3) = \{d^{3(2k)}e \mid k \geq 0\}$

$Ans(x_2,x_3) = \{d^{3(2k+1)}e \mid k \geq 0\}$

Just like the previous case, for the node x_3 we have $Ans(x_3,x_1) = Ans(x_3,x_2) = Ans(x_3,x_3) = \{no\}$ because $ASP(x_3,x_1) = ASP(x_3,x_2) = ASP(x_3,x_3) = \emptyset$

So we obtain the following language for the interpretation Σ_1:

$$\mathcal{L}(M,R) = \{d^{3k} \mid k \geq 1\} \cup \{d^{3i}e \mid i \geq 0\}.$$

We emphasize now an essential aspect of the representation proposed in this paper: the use of the set M and the definition of the partial operation $*$. In order to treat this aspect we consider the labelled graph drawn in Fig. 2.

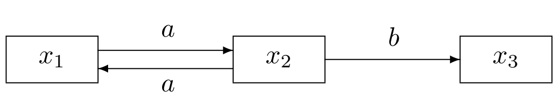

Figure 2. Labelled graph

We take $L_0 = \{a,b\}$, $T_0 = \{\rho_1,\rho_2\}$, where:

$\rho_1 = \{(x_1,x_2),(x_2,x_1)\}$,

$\rho_2 = \{(x_2,x_3)\}$.

Consider $u = prod_S$, where $S = \{x_1,x_2,x_3\}$.

We denote by $\mathcal{G}$ the labelled stratified graph generated by u. We obtain the new binary relations:

$\rho_3 = \{(x_1,x_1),(x_2,x_2)\}$ and

$\rho_4 = \{(x_1,x_3)\}$.

So we have: $T = \{\rho_1,\rho_2,\rho_3,\rho_4\}$.

$u(\rho_1,\rho_1) = \rho_3$, $u(\rho_1,\rho_2) = \rho_4$, $u(\rho_1,\rho_3) = \rho_1$,

$u(\rho_1,\rho_4) = \rho_2$, $u(\rho_3,\rho_1) = \rho_1$, $u(\rho_3,\rho_2) = \rho_2$,

$u(\rho_3,\rho_3) = \rho_3$, $u(\rho_3,\rho_4) = \rho_4$.

In Fig. 3 we observe the computation of the element D_n as calculated in [14]. Considering the values of the mapping f we obtain four containers of labels, each of them consisted of all the labels for ρ_1, ρ_2, ρ_3 and ρ_4 respectively. Each container has an infinite set of labels. In order to verify this fact we denote

$$\sigma(P,Q) = \{\sigma(u,v) \mid u \in P, v \in Q\}$$

and for each natural number n we take

$$\sigma_n(A,B) = \bigcup_{j \leq n} [\sigma(A_n,B_j) \cup \sigma(A_j,B_n)]$$

where A_j, B_j are subsets of L, A is the sequence $A_0, A_1, ...$ and B is the sequence $B_0, B_1, ...$ For every $j \geq 0$ and $i \in \{1,2,3,4\}$ we denote $D_j(\rho_i) = \{u \in D_j \mid f(u) = \rho_i\}$ and let $D(\rho_i)$ be the sequence $D_0(\rho_i), D_1(\rho_i), ...$

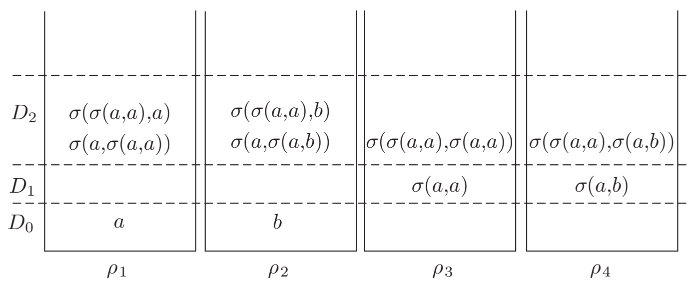

Figure 3. The infinite hierarchy of layers

We obtain the following equations:

$$\begin{cases}
D_{n+1}(\rho_1) = \sigma_n(D(\rho_1),D(\rho_3)) \cup \sigma_n(D(\rho_3),D(\rho_1)) \\
D_{n+1}(\rho_2) = \sigma_n(D(\rho_1),D(\rho_4)) \cup \sigma_n(D(\rho_3),D(\rho_2)) \\
D_{n+1}(\rho_3) = \sigma_n(D(\rho_1),D(\rho_1)) \cup \sigma_n(D(\rho_3),D(\rho_3)) \\
D_{n+1}(\rho_4) = \sigma_n(D(\rho_1),D(\rho_2)) \cup \sigma_n(D(\rho_3),D(\rho_4))
\end{cases} \tag{6}$$

We observe that $D_2(\rho_1)$, $D_2(\rho_2)$, $D_2(\rho_3)$ and $D_2(\rho_4)$ are nonempty sets. Based on (6) we can verify by induction that $D_n(\rho_1)$, $D_n(\rho_2)$, $D_n(\rho_3)$ and $D_n(\rho_4)$ are also nonempty sets for every $n \geq 3$. Thus we obtain an infinite hierarchy of layers for L.

We shall write $Ans(x,y) = \emptyset$ if $ASP_R(x,y) = \emptyset$.

Consider the following interpretation $\Sigma = (V^*, i, D, \mathcal{P})$, where

- V^* is the free monoid generated by $V = \{a_1,a_2,a_3\}$;
- $i : S \longrightarrow V^*$, $i(x_1) = a_1$, $i(x_2) = a_2$, $i(x_3) = a_3$;
- $D = (V^*, *)$ is a partial algebra, where $* : V^* \times V^* \longrightarrow V^*$ is defined as follows: for every natural number k we take:

$$(a_1^k a_2^k) * a_1 = a_1^{k+1} a_2^k \tag{7}$$

$$(a_1^{k+1}a_2^k) * a_2 = a_1^{k+1}a_2^{k+1} \quad (8)$$

$$(a_1^{k+1}a_2^k) * a_2a_3 = a_1^{k+1}a_2^{k+1}a_3^{k+1} \quad (9)$$

We observe that $*$ is in this case a partial operation.

○ The algorithms $Alg_a : V^* \times V^* \longrightarrow V^*$ and $Alg_b : V^* \times V^* \longrightarrow V^*$ are defined as follows:

$Alg_a(x, y) = x$;
$Alg_b(x, y) = xy$.

and take $\mathcal{P} = \{Alg_a, Alg_b\}$.

We choose the set $M = \{(x_1, x_3)\}$. We know that the pair (x_1, x_3) corresponds to the binary relation ρ_4 so we obtain the labels for the accepted structured paths from x_1 to x_3 in the fourth container which is infinite as we have seen. Therefore we have an infinity of accepted structured paths from x_1 to x_3.

Using the set R of restrictions we impose the restriction to use only the accepted structured paths of the following form: $([x_1, x_2, x_1, x_2, \ldots, x_1, x_2, x_3], \sigma(\omega_k, b))$, where the pair (x_1, x_2) appears k times and $\omega_k = \sigma(\sigma(\ldots\sigma(a, a), a)\ldots, a)$ contains $2(k-1)$ letters σ and $2k-1$ letters a.

So we obtain: $ASP_R(\mathcal{G}) = ASP_R(x_1, x_3) = \bigcup_{k\geq 1}\{([x_1, x_2, \ldots, x_1, x_2, x_3], \sigma(\omega_k, b))\}$.

Proposition 3.1: The language generated by the sets M and R is:

$$\mathcal{L}(M, R) = \{a_1^k a_2^k a_3^k\}_{k\geq 1}$$

Proof:

First we prove

$$\{a_1^k a_2^k a_3^k\}_{k\geq 1} \subseteq \mathcal{L}(M, R) \quad (10)$$

We have: $\mathcal{L}(M, R) = \underset{(x,y)\in M}{\cup} Ans(x, y)$, but $M = \{(x_1, x_3)\}$ so we have $\mathcal{L}(M, R) = Ans(x_1, x_3) = \{val_\Sigma(d) \mid d \in ASP_R(x_1, x_3)\}$

We denote by d_k the accepted structured path $([x_1, x_2, \ldots, x_1, x_2, x_3], \sigma(\omega_k, b))$, where the pair (x_1, x_2) appears k times and $\omega_k = \sigma(\sigma(\ldots\sigma(a, a), a)\ldots, a)$ has $2(k-1)$ letters σ and $2k-1$ letters a.

So $ASP_R(x_1, x_3) = \bigcup_{k\geq 1} d_k$ and $Ans(x_1, x_3) = \bigcup_{k\geq 1}\{val_\Sigma(d_k)\}$.

According to the definition of val_Σ we have $val_\Sigma(x(1; n+1), \sigma(v_1, v_2)) = val_\Sigma(x(1; , i), v_1) * val_\Sigma(x(i; n+1), v_2)$, where $i = ind([x_1, ..., x_{n+1}], \sigma(v_1, v_2))$ and $x(i; j) = [x_i, ..., x_j]$.

Therefore we have: $val_\Sigma(d_k) = val_\Sigma([x_1, x_2, \ldots, x_1, x_2], \omega_k) * val_\Sigma([x_2, x_3], b) = val_\Sigma([x_1, x_2, \ldots, x_1, x_2], \omega_k) * Alg_b(i(x_2), i(x_3)) = val_\Sigma([x_1, x_2, \ldots, x_1, x_2], \omega_k) * Alg_b(a_2, a_3) = val_\Sigma([x_1, x_2, \ldots, x_1, x_2], \omega_k) * a_2a_3$, for every $k \geq 1$.

We prove (10) by induction for k.

For $k = 1$ we have $d_1 = ([x_1, x_2, x_3], \sigma(a, b))$ and applying val_Σ to d_1 we obtain:
$val_\Sigma([x_1, x_2, x_3], \sigma(a, b)) = val_\Sigma([x_1, x_2], a) * a_2a_3 = Alg_a(i(x_1), i(x_2)) * a_2a_3 = Alg_a(a_1, a_2) * a_2a_3 = a_1 * a_2a_3 = a_1a_2a_3$ applying (9) therefore we have

$$a_1a_2a_3 \in Ans(x_1, x_3)$$

We suppose the property is true for k so for d_k we have

$$a_1^k a_2^k a_3^k \in Ans(x_1, x_3)$$

For $val_\Sigma(d_k)$ we have:
$val_\Sigma([x_1, x_2, \ldots, x_1, x_2, x_3], \sigma(\omega_k, b)) = (...(val_\Sigma([x_1, x_2], a) * val_\Sigma([x_2, x_1], a)) * val_\Sigma([x_1, x_2], a) * ... * val_\Sigma([x_1, x_2], a) * a_2a_3$, where $val_\Sigma([x_1, x_2], a)$ appears k times composed successively with $val_\Sigma([x_2, x_1], a)$ which appears $k-1$ times.

We applied (8) and (7) successively, one for every $*$ operation but the last one ($2(k-1)$ times) and we obtain $a_1^k a_2^{k-1}$ the computation for the k pairs (x_1, x_2), giving us the final result: $(a_1^k a_2^{k-1}) * a_2a_3$ which from (9) is $a_1^k a_2^k a_3^k$.

For $k+1$ we have d_{k+1} that has $k+1$ pairs (x_1, x_2). So when computing $val_\Sigma(d_{k+1})$ we obtain:
$val_\Sigma([x_1, x_2, \ldots, x_1, x_2, x_3], \sigma(\omega_{k+1}, b)) = (...(val_\Sigma([x_1, x_2], a) * val_\Sigma([x_2, x_1], a)) * val_\Sigma([x_1, x_2], a) * ... * val_\Sigma([x_1, x_2], a) * a_2a_3$, where $val_\Sigma([x_1, x_2], a)$ appears $k+1$ times composed successively with $val_\Sigma([x_2, x_1], a)$ which appears k times.

From step k we have the value for the first k pairs (x_1, x_2) and we add one more so we obtain:
$((a_1^k a_2^{k-1} * a_2) * a_1) * a_2a_3 \overset{(8)}{=} (a_1^k a_2^k * a_1) * a_2a_3 \overset{(7)}{=} a_1^{k+1} a_2^k * a_2a_3 \overset{(9)}{=} a_1^{k+1} a_2^{k+1} a_3^{k+1}$.

So we have

$$a_1^{k+1} a_2^{k+1} a_3^{k+1} \in Ans(x_1, x_3)$$

Therefore we have $\{a_1^k a_2^k a_3^k\}_{k\geq 1} \subseteq Ans(x_1, x_3) = \mathcal{L}(M, R)$.

Now we prove

$$\mathcal{L}(M, R) \subseteq \{a_1^k a_2^k a_3^k\}_{k\geq 1} \quad (11)$$

For every $k \geq 1$ there is $d_k \in ASP_R(x_1, x_3)$ such that $val_\Sigma(d_k) = a_1^k a_2^k a_3^k$.

So $\mathcal{L}(M, R) = Ans(x_1, x_3) = \bigcup_{k\geq 1}\{val_\Sigma(d_k)\} = \bigcup_{k\geq 1}\{a_1^k a_2^k a_3^k\} \subseteq \{a_1^k a_2^k a_3^k\}_{k\geq 1}$.

■

The language we obtained: $\{a_1^k a_2^k a_3^k\}_{k\geq 1}$ is a context-sensitive language, but is not a context free language in Chomsky hierarchy.

Because of the restriction we imposed this is the only language we can obtain from the computations on $ASP_R(\mathcal{G})$.

Let's now remove the restrictions that we have defined before so we have $ASP_R(\mathcal{G}) = ASP(\mathcal{G})$ and we will redefine the operation $*$ as follows: for every $i, j, k, l \geq 0$ and $m \geq 1$ we take

$$a_1^i a_2^j * a_1^k a_2^l = a_1^{i+k} a_2^{j+l} \quad (12)$$

$$a_1^i a_2^j * a_1^k a_2^l a_3^m = a_1^{i+k} a_2^{j+l} a_3^{j+l} \quad (13)$$

Because we have removed the restrictions we will have even more accepted structured paths than before in $ASP_R(\mathcal{G})$.

If we choose $M_1 = \{(x_1, x_3)\}$ we will also obtain $\mathcal{L}(M_1, R) = \{a_1^k a_2^k a_3^k\}_{k\geq 1}$, but every word in the language will be generated by more accepted structured paths. For example if we consider the path: $[x_1, x_2, x_1, x_2, x_3]$ we obtain the following accepted structured paths according to the labels we have in the container ρ_4:

1) $([x_1,x_2,x_1,x_2,x_3],\sigma(\sigma(\sigma(a,a),a),b))$
2) $([x_1,x_2,x_1,x_2,x_3],\sigma(\sigma(a,a),\sigma(a,b)))$
3) $([x_1,x_2,x_1,x_2,x_3],\sigma(\sigma(a,\sigma(a,a)),b))$
4) $([x_1,x_2,x_1,x_2,x_3],\sigma(a,\sigma(\sigma(a,a),b)))$
5) $([x_1,x_2,x_1,x_2,x_3],\sigma(a,\sigma(a,\sigma(a,b))))$

When computing each of them we obtain the same result, using the new defined operation $*$.

1) $val_\Sigma([x_1,x_2,x_1,x_2,x_3],\sigma(\sigma(\sigma(a,a),a),b)) = val_\Sigma([x_1,x_2,x_1,x_2],\sigma(\sigma(a,a),a))*val_\Sigma([x_2,x_3],b) = (val_\Sigma([x_1,x_2,x_1],\sigma(a,a))*val_\Sigma([x_1,x_2],a))* Alg_b(i(x_2),i(x_3)) = ((val_\Sigma([x_1,x_2],a)* val_\Sigma([x_2,x_1],a))*Alg_a(i(x_1),i(x_2)))*Alg_b(a_2,a_3) = ((Alg_a(i(x_1),i(x_2))*Alg_a(i(x_2),i(x_1)))* Alg_a(a_1,a_2))*a_2a_3 = ((Alg_a(a_1,a_2)* Alg_a(a_2,a_1))*a_1)*a_2a_3 = ((a_1*a_2)*a_1)*a_2a_3 \stackrel{(12)}{=} (a_1a_2*a_1)*a_2a_3 \stackrel{(12)}{=} a_1^2a_2*a_2a_3 \stackrel{(13)}{=} a_1^2a_2^2a_3^2.$
2) $val_\Sigma([x_1,x_2,x_1,x_2,x_3],\sigma(\sigma(a,a),\sigma(a,b))) = val_\Sigma([x_1,x_2,x_1],\sigma(a,a))*val_\Sigma([x_1,x_2,x_3],\sigma(a,b)) = (val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1],a))* (val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_3],b)) = (Alg_a(i(x_1),i(x_2))*Alg_a(i(x_2),i(x_1)))* (Alg_a(i(x_1),i(x_2))*Alg_b(i(x_2),i(x_3))) = (Alg_a(a_1,a_2)*Alg_a(a_2,a_1))*(Alg_a(a_1,a_2)* Alg_b(a_2,a_3)) = (a_1*a_2)*(a_1*a_2a_3) \stackrel{(12)(13)}{=} a_1a_2*a_1a_2a_3 \stackrel{(13)}{=} a_1^2a_2^2a_3^2.$
3) $val_\Sigma([x_1,x_2,x_1,x_2,x_3],\sigma(\sigma(a,\sigma(a,a)),b)) = val_\Sigma([x_1,x_2,x_1,x_2],\sigma(a,\sigma(a,a)))*val_\Sigma([x_2,x_3],b) = (val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1,x_2],\sigma(a,a)))* Alg_b(i(x_2),i(x_3)) = (Alg_a(i(x_1),i(x_2))* (val_\Sigma([x_2,x_1],a)*val_\Sigma([x_1,x_2],a)))*Alg_b(a_2,a_3) = (Alg_a(a_1,a_2)*(Alg_a(i(x_2),i(x_1))* Alg_a(i(x_1),i(x_2))))*a_2a_3 = (a_1*(Alg_a(a_2,a_1)* Alg_a(a_1,a_2)))*a_2a_3 = (a_1*(a_2*a_1))*a_2a_3 \stackrel{(12)}{=} (a_1*a_1a_2)*a_2a_3 \stackrel{(12)}{=} a_1^2a_2*a_2a_3 \stackrel{(13)}{=} a_1^2a_2^2a_3^2.$
4) $val_\Sigma([x_1,x_2,x_1,x_2,x_3],\sigma(a,\sigma(\sigma(a,a),b))) = val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1,x_2,x_3],\sigma(\sigma(a,a),b)) = Alg_a(i(x_1),i(x_2))*(val_\Sigma([x_2,x_1,x_2],\sigma(a,a))* val_\Sigma([x_2,x_3],b)) = Alg_a(a_1,a_2)*((val_\Sigma([x_2,x_1],a)* val_\Sigma([x_1,x_2],a))*Alg_b(i(x_2),i(x_3))) = a_1*((Alg_a(i(x_2),i(x_1))*Alg_a(i(x_1),i(x_2)))* Alg_b(a_2,a_3)) = a_1*((Alg_a(a_2,a_1)*Alg_a(a_1,a_2))* a_2a_3) = a_1*((a_2*a_1)*a_2a_3) \stackrel{(12)}{=} a_1*(a_1a_2*a_2a_3) \stackrel{(13)}{=} a_1*a_1a_2^2a_3^2 \stackrel{(13)}{=} a_1^2a_2^2a_3^2.$
5) $val_\Sigma([x_1,x_2,x_1,x_2,x_3],\sigma(a,\sigma(a,\sigma(a,b)))) = val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1,x_2,x_3],\sigma(a,\sigma(a,b))) = Alg_a(i(x_1),i(x_2))*(val_\Sigma([x_2,x_1],a)* val_\Sigma([x_1,x_2,x_3],\sigma(a,b))) = Alg_a(a_1,a_2)*(Alg_a(i(x_2),i(x_1))*(val_\Sigma([x_1,x_2],a)* val_\Sigma([x_2,x_3],b))) = a_1*(Alg_a(a_2,a_1)* (Alg_a(i(x_1),i(x_2))*Alg_b(i(x_2),i(x_3))))) = a_1*(a_2*(Alg_a(a_1,a_2)*Alg_b(a_2,a_3))) = a_1*(a_2*(a_1*a_2a_3)) \stackrel{(13)}{=} a_1*(a_2*a_1a_2a_3) \stackrel{(13)}{=} a_1*a_1a_2^2a_3^2 \stackrel{(13)}{=} a_1^2a_2^2a_3^2.$

Next we show what results we obtain if we choose different sets M.

For the set $M_2=\{(x_2,x_3)\}$ we obtain accepted structured paths with labels from the container corresponding to ρ_2 and we compute one for the first 3 paths from x_2 to x_3:

- $val_\Sigma([x_2,x_3],b) = Alg_b(i(x_2),i(x_3)) = Alg_b(a_2,a_3) = a_2a_3.$
- $val_\Sigma([x_2,x_1,x_2,x_3],\sigma(a,\sigma(a,b))) = val_\Sigma([x_2,x_1],a)* val_\Sigma([x_1,x_2,x_3],\sigma(a,b)) = Alg_a(i(x_2),i(x_1))* (val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_3],b)) = Alg_a(a_2,a_1)*(Alg_a(i(x_1),i(x_2))*Alg_b(i(x_2),i(x_3))) = a_2*(Alg_a(a_1,a_2)*Alg_b(a_2,a_3)) = a_2*(a_1*a_2a_3) \stackrel{(13)}{=} a_2*a_1a_2a_3 \stackrel{(13)}{=} a_1a_2^2a_3^2.$
- $val_\Sigma([x_2,x_1,x_2,x_1,x_2,x_3],\sigma(\sigma(a,a),\sigma(a,\sigma(a,b)))) = (val_\Sigma([x_2,x_1],a)*val_\Sigma([x_1,x_2],a))*(val_\Sigma([x_2,x_1],a)* val_\Sigma([x_1,x_2,x_3],\sigma(a,b))) = (Alg_a(i(x_2),i(x_1))* Alg_a(i(x_1),i(x_2)))*(Alg_a(i(x_2),i(x_1))* (val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_3],b))) = (Alg_a(a_2,a_1)*Alg_a(a_1,a_2))*(Alg_a(a_2,a_1)* (Alg_a(i(x_1),i(x_2))*Alg_b(i(x_2),i(x_3)))) = (a_2*a_1)*(a_2*(Alg_a(a_1,a_2)*Alg_b(a_2,a_3))) \stackrel{(12)}{=} a_1a_2*(a_2*(a_1*a_2a_3)) \stackrel{(13)}{=} a_1a_2*(a_2*a_1a_2a_3) \stackrel{(13)}{=} a_1a_2*a_1a_2^2a_3^2 \stackrel{(13)}{=} a_1^2a_2^3a_3^3.$

So continuing the computations for any accepted structured path from x_2 to x_3 we obtain the language: $\mathcal{L}(M_2,R) = \{a_1^{k-1}a_2^ka_3^k\}_{k\geq 1}$.

For the set $M_3=\{(x_1,x_2)\}$ we obtain:

- $val_\Sigma([x_1,x_2],a) = Alg_a(i(x_1),i(x_2)) = Alg_a(a_1,a_2) = a_1.$
- $val_\Sigma([x_1,x_2,x_1,x_2],\sigma(a,\sigma(a,a))) = val_\Sigma([x_1,x_2],a)* val_\Sigma([x_2,x_1,x_2],\sigma(a,a)) = Alg_a(i(x_1),i(x_2))* (val_\Sigma([x_2,x_1],a)*val_\Sigma([x_1,x_2],a)) = Alg_a(a_1,a_2)*(Alg_a(i(x_2),i(x_1))*Alg_a(i(x_1),i(x_2))) = a_1*(Alg_a(a_2,a_1)*Alg_a(a_1,a_2)) = a_1*(a_2*a_1) \stackrel{(12)}{=} a_1*a_1a_2 \stackrel{(12)}{=} a_1^2a_2.$
- $val_\Sigma([x_1,x_2,x_1,x_2,x_1,x_2],\sigma(\sigma(\sigma(a,a),\sigma(a,a)),a)) = ((val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1],a))* (val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1],a)))* val_\Sigma([x_1,x_2],a) = ((Alg_a(i(x_1),i(x_2))* Alg_a(i(x_2),i(x_1)))*(Alg_a(i(x_1),i(x_2))* Alg_a(i(x_2),i(x_1))))*Alg_a(i(x_1),i(x_2)) = ((Alg_a(a_1,a_2)*Alg_a(a_2,a_1))*(Alg_a(a_1,a_2)* Alg_a(a_2,a_1)))*Alg_a(a_1,a_2) = ((a_1*a_2)*(a_1*a_2))* a_1 \stackrel{(12)}{=} (a_1a_2*a_1a_2)*a_1 \stackrel{(12)}{=} a_1^2a_2^2*a_1 \stackrel{(12)}{=} a_1^3a_2^2.$

We obtain the language: $\mathcal{L}(M_3,R) = \{a_1^ka_2^{k-1}\}_{k\geq 1}$.

For the set $M_4=\{(x_2,x_1)\}$ we obtain:

- $val_\Sigma([x_2,x_1],a) = Alg_a(i(x_2),i(x_1)) = Alg_a(a_2,a_1) = a_2.$
- $val_\Sigma([x_2,x_1,x_2,x_1],\sigma(\sigma(a,a),a)) = val_\Sigma([x_2,x_1,x_2],\sigma(a,a))*val_\Sigma([x_2,x_1],a) = (val_\Sigma([x_2,x_1],a)*val_\Sigma([x_1,x_2],a))* Alg_a(i(x_2),i(x_1)) = (Alg_a(i(x_2),i(x_1))*Alg_a(i(x_1),i(x_2)))*Alg_a(a_2,a_1) = (Alg_a(a_2,a_1)*Alg_a(a_1,a_2))*a_2 = (a_2*a_1)*a_2 \stackrel{(12)}{=} a_1a_2*a_2 \stackrel{(12)}{=} a_1a_2^2.$

- $val_\Sigma([x_2,x_1,x_2,x_1,x_2,x_1],\sigma(\sigma(a,a),\sigma(a,\sigma(a,a)))) = (val_\Sigma([x_2,x_1],a)*val_\Sigma([x_1,x_2],a))*(val_\Sigma([x_2,x_1],a)* val_\Sigma([x_1,x_2,x_1],\sigma(a,a))) = (Alg_a(i(x_2),i(x_1))* Alg_a(i(x_1),i(x_2)))*(Alg_a(i(x_2),i(x_1))* (val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1],a))) = (Alg_a(a_2,a_1)*Alg_a(a_1,a_2))*(Alg_a(a_2,a_1)* (Alg_a(i(x_1),i(x_2))*Alg_a(i(x_2),i(x_1)))) = (a_2*a_1)*(a_2*(Alg_a(a_1,a_2)*Alg_a(a_2,a_1))) \overset{(12)}{=} a_1a_2*(a_2*(a_1*a_2)) \overset{(12)}{=} a_1a_2*(a_2*a_1a_2) \overset{(12)}{=} a_1a_2*a_1a_2^2 \overset{(12)}{=} a_1^2a_2^3$.

We obtain the language: $\mathcal{L}(M_4,R) = \{a_1^{k-1}a_2^k\}_{k\geq 1}$.
For the set $M_5 = \{(x_1,x_1)\}$ we obtain:

- $val_\Sigma([x_1,x_2,x_1],\sigma(a,a)) = val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1],a) = Alg_a(i(x_1),i(x_2))*Alg_a(i(x_2),i(x_1)) = Alg_a(a_1,a_2)*Alg_a(a_2,a_1) = a_1*a_2 \overset{(12)}{=} a_1a_2$.
- $val_\Sigma([x_1,x_2,x_1,x_2,x_1],\sigma(\sigma(a,a),\sigma(a,a))) = val_\Sigma([x_1,x_2,x_1],\sigma(a,a))*val_\Sigma([x_1,x_2,x_1],\sigma(a,a)) = (val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1],a))*(val_\Sigma([x_1,x_2],a)* val_\Sigma([x_2,x_1],a)) = (Alg_a(i(x_1),i(x_2))* Alg_a(i(x_2),i(x_1)))*(Alg_a(i(x_1),i(x_2))* Alg_a(i(x_2),i(x_1))) = (Alg_a(a_1,a_2)*Alg_a(a_2,a_1))* (Alg_a(a_1,a_2)*Alg_a(a_2,a_1)) = (a_1*a_2)*(a_1*a_2) \overset{(12)}{=} a_1a_2*a_1a_2 \overset{(12)}{=} a_1^2a_2^2$.
- $val_\Sigma([x_1,x_2,x_1,x_2,x_1,x_2,x_1],\sigma(\sigma(\sigma(a,a), \sigma(a,a)),\sigma(a,a))) = val_\Sigma([x_1,x_2,x_1,x_2,x_1],\sigma(\sigma(a,a), \sigma(a,a)))*val_\Sigma([x_1,x_2,x_1],\sigma(a,a)) = (val_\Sigma([x_1,x_2,x_1],\sigma(a,a))*val_\Sigma([x_1,x_2,x_1],\sigma(a,a)))* (val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1],a)) = ((val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1],a))* (val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1],a)))* val_\Sigma([x_1,x_2],a)*val_\Sigma([x_2,x_1],a) = (Alg_a(a_1,a_2)*Alg_a(a_2,a_1))*(Alg_a(a_1,a_2)* Alg_a(a_2,a_1))*(Alg_a(a_1,a_2)*Alg_a(a_2,a_1)) = ((a_1*a_2)*(a_1*a_2))*(a_1*a_2) \overset{(12)}{=} (a_1a_2*a_1a_2)*a_1a_2 \overset{(12)}{=} a_1^2a_2^2*a_1a_2 \overset{(12)}{=} a_1^3a_2^3$.

We obtain the language: $\mathcal{L}(M_5,R) = \{a_1^ka_2^k\}_{k\geq 1}$.
For the set $M_6 = \{(x_2,x_2)\}$ we obtain:

- $val_\Sigma([x_2,x_1,x_2],\sigma(a,a)) = val_\Sigma([x_2,x_1],a)*val_\Sigma([x_1,x_2],a) = Alg_a(i(x_2),i(x_1))*Alg_a(i(x_1),i(x_2)) = Alg_a(a_2,a_1)*Alg_a(a_1,a_2) = a_2*a_1 \overset{(12)}{=} a_1a_2$.
- $val_\Sigma([x_2,x_1,x_2,x_1,x_2],\sigma(\sigma(\sigma(a,a),a),a)) = val_\Sigma([x_2,x_1,x_2,x_1,],\sigma(\sigma(a,a),a))*val_\Sigma([x_1,x_2],a) = ((val_\Sigma([x_2,x_1],a)*val_\Sigma([x_1,x_2],a))* (val_\Sigma([x_2,x_1],a))*val_\Sigma([x_1,x_2],a) = ((Alg_a(i(x_2),i(x_1))*Alg_a(i(x_1),i(x_2)))* Alg_a(i(x_2),i(x_1))*Alg_a(i(x_1),i(x_2)) = ((Alg_a(a_2,a_1)*Alg_a(a_1,a_2))*Alg_a(a_2,a_1))* Alg_a(a_1,a_2) = ((a_2*a_1)*a_2)*a_1 \overset{(12)}{=} (a_1a_2*a_2)*a_1 \overset{(12)}{=} a_1a_2^2*a_1 \overset{(12)}{=} a_1^2a_2^2$.
- $val_\Sigma([x_2,x_1,x_2,x_1,x_2,x_1,x_2],\sigma(\sigma(\sigma(a,a),\sigma(a,a)), \sigma(a,a))) = val_\Sigma([x_2,x_1,x_2,x_1,x_2],\sigma(\sigma(a,a), \sigma(a,a)))*val_\Sigma([x_2,x_1,x_2],\sigma(a,a)) = (val_\Sigma([x_2,x_1,x_2],\sigma(a,a))*val_\Sigma([x_2,x_1,x_2],\sigma(a,a)))* (val_\Sigma([x_2,x_1],a)*val_\Sigma([x_1,x_2],a)) = ((val_\Sigma([x_2,x_1],a)*val_\Sigma([x_1,x_2],a))* (val_\Sigma([x_2,x_1],a)*val_\Sigma([x_1,x_2],a)))*val_\Sigma([x_2,x_1],a)* val_\Sigma([x_1,x_2],a) = (Alg_a(a_2,a_1)*Alg_a(a_1,a_2))* (Alg_a(a_2,a_1)*Alg_a(a_1,a_2))*(Alg_a(a_2,a_1)* Alg_a(a_1,a_2)) = ((a_2*a_1)*(a_2*a_1))*(a_2*a_1) \overset{(12)}{=} (a_1a_2*a_1a_2)*a_1a_2 \overset{(12)}{=} a_1^2a_2^2*a_1a_2 \overset{(12)}{=} a_1^3a_2^3$.

We obtain the language: $\mathcal{L}(M_6,R) = \{a_1^ka_2^k\}_{k\geq 1}$.

We can see that $\mathcal{L}(M_5,R) = \mathcal{L}(M_6,R)$ so we can write: $\mathcal{L}(M_7,R) = \{a_1^ka_2^k\}_{k\geq 1}$, where $M_7 = \{(x_1,x_1),(x_2,x_2)\}$. This is a context-free language but not a regular language, according to *Pumping Lemma* for regular languages.

IV. Conclusions and open problems

In this paper we present a new application of the stratified graphs $\mathcal{G}$. We showed that by a specific interpretation $\mathcal{I}$ of $\mathcal{G}$ we obtain a formal language. Also by adding the restriction set R we can limit the computations to a certain type of accepted structured paths and use only those to obtain a certain language.

We relieved that these structure can generate regular languages, context-sensitive languages which are not context-free languages and also context-free languages that are not regular languages and even more they can all be generated using the same labelled stratified graph. This study can be continued and we relieve here the following open problems:

1) Study the family of languages generated by labelled stratified graphs in comparison with the Chomsky hierarchy of formal languages.
2) There are several syntactical mechanisms to model the natural languages (for example, Recursive Transition Networks, Augmented Transition Networks etc). Try to apply the ideas presented in this paper to obtain subsets of natural languages by means of labelled stratified graphs.
3) Characterize the algebraic properties of formal languages generated by the same labelled stratified graph, but using various interpretations.
4) Study the change of $\mathcal{L}(M,R)$ for various choices of R and M.
5) The most interesting languages are the infinite ones. Study the infiniteness of the languages represented by stratified graphs by means of the containers of such a structure (defined as in [14]).
6) The Lindenmayer systems can also generate formal languages. Study the family of the languages generated by stratified graphs in comparison with the languages generated by the Lindenmayer systems.
7) Study the problem described in this paper with another kind of interpretation. Namely, for each $x \in S$ the element $i(x)$ is the language generated by some formal grammar. The relations between the nodes of a stratified graph, as well as the operation $*$, are defined in this case as a binary operation between formal languages. In this manner we can combine the languages generated by distinct kinds of grammar or we can combine a

formal language generated by a grammar with the formal language generated by a Lindenmayer system.

REFERENCES

[1] P.Prusinkiewicz, A.Lindenmayer, The Algorithmic Beauty of Plants. Springer-Verlag:New York, 1990

[2] V. Boicescu, A. Filipoiu, G.Georgescu, S.Rudeanu, "Lukasiewicz-Moisil Algebras", *Annals of Discrete Mathematics* 49, sNorth-Holland, 1991

[3] Way, E.C., Knowledge Representation and Metaphor, Kluwer Academic Publisher, Studies in cognitive systems, Vol. 7, 1991.

[4] N. Ţăndăreanu, Knowledge Bases with Output, Knowledge and Information Systems, 2(4), 2000, 438-460

[5] John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison Wesley, (2000)

[6] N. Ţăndăreanu, Proving the existence of labeled Stratified Graphs, *Annals of the University of Craiova, Mathematics and Computer Science Series*, Vol. XXVII, (2000), 81-92

[7] N. Ţăndăreanu, M. Ghindeanu, Image Synthesis from Natural Language Description, Research Notes in Artificial Intelligence and Digital Communications, Vol.103, 3^{nd} *Romanian Conference on Artificial Intelligence and Digital Communications*, Craiova, June 2003, p.82-96

[8] N. Ţăndăreanu, Collaborations between distinguished representatives for labeled stratified graphs, Annals of the University of Craiova, Mathematics and Computer Science Series, Vol. 30(2), 2003, p.184-192

[9] N. Ţăndăreanu, Intuitive Aspects of the Semantic Computations in KBO, *Research Notes in Artificial Intelligence and Digital Communications 101*, 1^{st} National Conference of RCAI, p.1-8, June 2001

[10] N. Ţăndăreanu, Knowledge Representation by Labeled Stratified Graphs, The 8th World Multi-Conference on Systemics, Cybernetics and Informatics, July 18-21, 2004, Orlando, Florida, USA, Vol. V: Computer Science and Engineering, p.345-350, 2004

[11] N. Ţăndăreanu and M. Ghindeanu, Hierarchical Reasoning Based on Stratified Graphs. Application in Image Synthesis., *15th International Workshop on Database and Expert Systems Applications, Proceedings of DEXA2004, Zaragoza, IEEE Computer Society, Los Alamitos California*, p.498-502, 2004

[12] N. Ţăndăreanu, Distinguished Representatives for Equivalent Labeled Stratified Graphs and Applications, *Discrete Applied Mathematics*, Vol. 144/1-2, p.183-208, 2004

[13] N. Ţăndăreanu, An Overview of Stratified Graphs and Their Applications, *Research Notes in Artificial Intelligence and Digital Communications*, Vol.105, 5th International Conference on Artificial Intelligence and Digital Communications, (2005), 111-120

[14] N. Ţăndăreanu, A recursive method to compute the layers of a stratified graph, *Journal of Multi-Valued Logic and Soft Computing*, Vol. 12, p.355-363, 2006