Right-Linear Languages Generated in Systems of Knowledge Representation based on LSG

Daniela Dănciulescu
University of Craiova,
Alexandru Ioan Cuza 13, Craiova 200585, Romania
danadanciulescu@gmail.com

Mihaela Colhon
University of Craiova,
Alexandru Ioan Cuza 13, Craiova 200585, Romania
mcolhon@inf.ucv.ro

Gheorghe Grigoraş
Alexandru Ioan Cuza University of Iaşi,
Bulevardul Carol I 11, Iaşi 700506, Romania
grigoras@info.uaic.ro

Abstract

In Tudor (Preda) (2010) a method for formal languages generation based on labeled stratified graph representations is sketched. The author proves that the considered method can generate regular languages and context-sensitive languages by considering an exemplification of the proposed method for a particular regular language and another one for a particular context-sensitive language. At the end of the study, the author highlights some open problems for future research among which we remind:
(1) The study of the language families that can be generated by means of these structures;
(2) The study of the infiniteness of the languages that can be represented in stratified graphs.
In this paper, we extend the method presented in Tudor (Preda)(2010), by considering the stratified graph formalism in a system of knowledge representation and reasoning. More precisely, we propose a method that can be applied for generating any Right Linear Language construction. Our method is proved and exemplified in several cases.

Keywords: graph-based representation and reasoning, formal language generation, right-linear language generation, accepted structured paths, stratified graph.

1. Introduction

A non-empty and finite set of symbols noted with Σ is considered in what follows the alphabet set. A word over an alphabet is a sequence of symbols $\omega = \alpha_1 \dots \alpha_n$, where $\alpha_i \in \Sigma, i = \overline{1,n}$. The length of a word will be denoted by $|\omega|$. Obviously, for $\omega = \alpha_1 \dots \alpha_n$, we have $|\omega| = n$ and $|\lambda| = 0$, where for λ we mark an empty word. The set of all nonempty words over an alphabet is denoted (instead of the set) Σ^+. If we want to consider the set of all the words over an alphabet (including the empty word) we denote $\Sigma^* = \Sigma^+ \cup \{\lambda\}$.

For two words: $\omega_1 = \alpha_1 \dots \alpha_n$ and $\omega_2 = \beta_1 \dots \beta_m$, we note by $\bullet$ the operation of string concatenation. We will have (1).

$$\omega_1 \bullet \omega_2 = \alpha_1 \dots \alpha_n \beta_1 \dots \beta_m \qquad (1)$$

This operation is not commutative, that is(2), but it is associative (3) with λ the neutral element (4).

$$\omega_1 \bullet \omega_2 \neq \omega_2 \bullet \omega_1 \qquad (2)$$
$$(\omega_1 \bullet \omega_2) \bullet \omega_3 = \omega_1 \bullet (\omega_2 \bullet \omega_3) \qquad (3)$$
$$\omega \bullet \lambda = \lambda \bullet \omega = \omega \qquad (4)$$

We obtain that $(\Sigma^{+}, \bullet)$ is a semi-group and $(\Sigma^{*}, \bullet)$ is a free monoid generated by Σ.

If we take $\omega_1, \omega_2 \in \Sigma^*$ we consider:

- ω_2 a sub-word of ω_1 if there are $\beta_1, \beta_2 \in \Sigma^*$ such that $\omega_1 = \beta_1\omega_2\beta_2$;
- ω_2 is a prefix of ω_1 if $\omega_1 = \omega_2\beta$ and $\beta \in \Sigma^*$;
- ω_2 is a suffix of ω_1 if $\omega_1 = \beta\omega_2$ and $\beta \in \Sigma^*$.

A language L over an alphabet Σ is defined as a subset of Σ^*, that is (5). Due to the fact that Σ^* is an infinite set, we can have the language (5) - an empty set, a finite or an infinite set.

We will note by $\omega = \alpha^n$ the word (6), obviously $\lambda = \alpha^0$.

$$L \subseteq \Sigma^* \quad (5)$$
$$\omega = \underbrace{\alpha \ldots \alpha}_{n-times} . \quad (6)$$

A grammar is considered as a tuple (7) such that (8), for Σ - the set of terminal symbols and N - the set of non-terminal symbols. S_0 is called the initial symbol and P is the set of grammar productions (rules).

$$G = (N, \Sigma, S_0, P) \quad (7)$$
$$\Sigma \cap N = \varnothing \quad (8)$$

The elements of "P" are of the form (9).

$$(\omega_i \rightarrow \omega_j) \in P \quad (9)$$

Where (10), such that ω_i contains at least a symbol from N and (11). In the rest of the paper, we call the *rules of the grammar G* the elements (12).A *direct derivation* in the grammar G is defined over $(N \cup \Sigma)^*$ using the binary relation (13).

$$\omega_i \in (N \cup \Sigma)^+ \quad (10)$$
$$\omega_j \in (N \cup \Sigma)^* \quad (11)$$
$$(\omega_i \rightarrow \omega_j) \in \mathrm{P} \quad (12)$$
$$\Rightarrow \text{ as: } \omega_i p \omega_j \Rightarrow \omega_i q \omega_j \text{ if } \omega_i, \omega_j,$$
$$p, q \in (N \cup \Sigma)^* \text{ and } \exists (p \rightarrow q) \in P \quad (13)$$

By $\Rightarrow^*$ we will note the reflexive and transitive closure of the relation $\Rightarrow$ and by $\Rightarrow^+$ the transitive closure of the same relation. Using these notations, the language generated by the grammar G is considered (14).

$$L(G) = \{\omega \in \Sigma^* \mid S_0 \Rightarrow^+ \omega\} \quad (14)$$

In this paper, we extend the method presented in Tudor (Preda)(2010), by considering the stratified graph formalism in a system for knowledge representation and reasoning. More precisely, we propose a method that can be applied for generating any Right Linear Language construction. Our method is proved and exemplified in several cases.

The paper is organized as follows: the second section provides the theoretical background behind the presented study. Section three presents the manner in which the language generation mechanism is designed by means of a system of knowledge representation and reasoning. The final section draws the concluding ideas and our future study.

2. System of knowledge representation based on stratified graph for formal language generation

We consider two finite sets S and L_0 such that (15), for S - a set of nodes labels and L_0 - a set of arcs labels. We note by (16) a non-empty set of binary relations on S and by f_0 a surjective function where (17). Following the definitions and the notations firstly introduced by Ţăndăreanu (Ţăndăreanu, 2000; Ţăndăreanu, 2003), the system (18) will denote a labeled graph structure.

$$S \setminus L_0 = \varnothing \quad (15)$$
$$T_0 \subseteq 2^{S\times S} \quad (16)$$
$$f_0\colon L_0 \to T_0 \quad (17)$$
$$G_0 = (S, L_0, T_0, f_0) \quad (18)$$

By $STR(G_0)$ we note the set of the structured paths of the graph G_0, this concept being defined (Dănciulescu and Ţăndăreanu, 2013) as follows:

- for every $a \in L_0$ and for every $x, y \in S$ such that (19), we have (20);

$$(x, y) \in f_0(a) \quad (19)$$
$$([x, y], a) \in STR(G_0) \quad (20)$$

- if (21) and (22) then (23).

$$([x_1, \ldots, x_n], u) \in STR(G_0) \quad (21)$$
$$([y_1, \ldots, y_m], v) \in STR(G_0) \quad (22)$$
$$([x_1, \ldots, x_n, y_1, \ldots, y_m], [u, v]) \in STR(G_0) \quad (23)$$

We take the projection of $STR(G_0)$ on the second axis (24).

$$STR_2(G_0) = \{u \mid \exists([x_1, \ldots, x_n], u) \in STR(G_0), n \geq 2\} \quad (24)$$

If we take σ_T the composition operation of binary relations from the set $2^{S\times S}$ defined as (25).

$$\sigma_T(\rho_1, \rho_2) = \{(x, z) \in 2^{S\times S} \mid \exists y \in S\colon (x, y) \in \rho_1, (y, z) \in \rho_2 \} \quad (25)$$

The pair $(2^{S\times S}, \sigma_T)$ becomes a partial algebra. Following (Ţăndăreanu, 2000), by the set (26) we will note the *closure* of the set T_0 in the algebra $(2^{S\times S}, \sigma_T)$.

$$\mathrm{T} = \mathrm{Cl}\sigma_T(T_0) \quad (26)$$

For σ_L - a binary relation symbol, we consider L the smallest set satisfying the following notations (Ţăndăreanu, 2000):

1) $L_0 \subseteq L$;
2) if (27) then (28).

$$a, b \in L_0 \quad (27)$$
$$\sigma_L(a, b) \in L \quad (28)$$

The pair (L, σ_L) also defines a *partial algebra*. Following Ţăndăreanu (Ţăndăreanu, 2000; Ţăndăreanu, 2003) we take (29).

$$f\colon (L, \sigma_L) \to (2^{S\times S}, \sigma_T) \quad (29)$$

A *morphism of partial algebras* such that (30) and if (31), then (32) (see Figure 1). We obtain $f(L) = T$ which means that "*for every element of L the associated element of T is computed by the morphism f*" (Ţăndăreanu, 2000).

$$a \in L_0, f(a) = f_0(a) \tag{30}$$
$$f(u), f(v)) \in dom(\sigma_T) \tag{31}$$
$$(u, v) \in dom(\sigma_L) \tag{32}$$

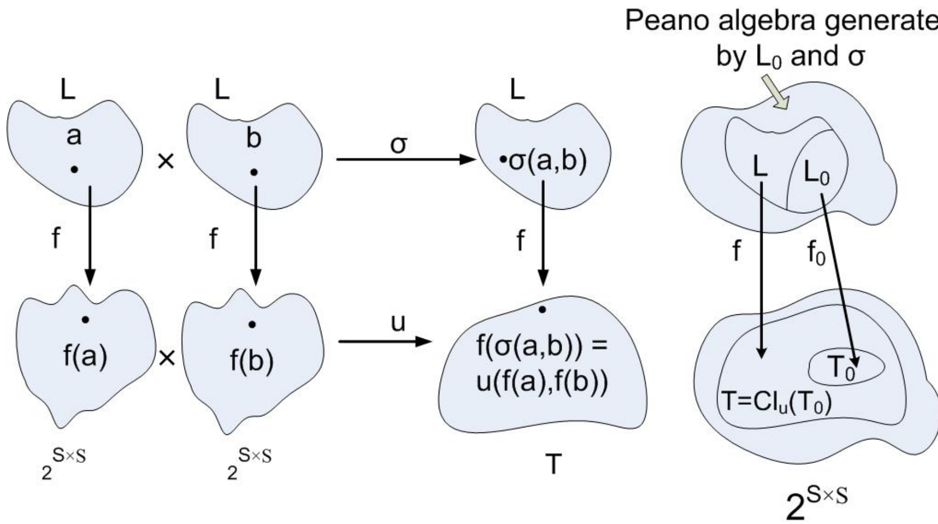

Figure 1. The graphical representation of the morphism f: (L, σ_L) → ($2^{S\times S}$, σ_T) (Ţăndăreanu, 2000)

From (Ţăndăreanu, 2000) for every (33) and for each (T, σ_T) we can construct a *Stratified Graph* (shortly, *SG*) over G_0: (34).

$$G_0 = (S, L_0, T_0, f_0) \tag{33}$$
$$\mathcal{G} = (G_0, L, T, \sigma_T , f) \tag{34}$$

The elements of *L* represent the labels for the binary relations of *T* defined in the stratified graph structure by means of σ_T - the composition operation of binary relations.

For each (35), we define *trace*(*u*) as the list of symbols from L_0, upon which the label *u* was made up, by considering the symbols of L_0 in the exactly the same order they appear in the label *u*.

$$u \in L \tag{35}$$

We have:

- if $u \in L_0$ then $trace(u) = [u]$
- if (36) with $a, b \in L_0$ then $trace(u) = [a, b]$
- if $u = \sigma_L(u_1, u_2)$ then $trace(u) = trace(\underline{u}_1) \cup trace(u_2)$,for$\cup$ we note the reunion of lists the operation (37).

$$u = \sigma_L(a, b) \tag{36}$$
$$[x_1, \ldots, x_n] \cup [y_1, \ldots, y_m] = [x_1, \ldots, x_n, y_1, \ldots, y_m]. \tag{37}$$

As it is considered in the literature, a formal language is a set of words which are represented by finite strings of letters, symbols or tokens. The set of all the letters or symbols upon which the words are made of is called the alphabet of the formal language. Usually, a formal language is defined by means of some formation rules or a "formal grammar" (Dănciulescu, 2015).

An interpretation for a stratified graph provides a representation in a domain of knowledge. In the presented approach, we take the domain of knowledge to be a particular formal language. In what follows we will note by V the alphabet of the generated formal language (Dănciulescu, 2015).

Therefore, let us consider the stratified graph (38) over a labeled graph (39). We defined the interpretation of $\mathcal{G}$ as a tuple: (40) (Dănciulescu, 2015).

$$\mathcal{G} = (G_0, L, T, u, f) \quad (38)$$
$$G_0 = (S_0, L_0, T_0, f_0) \quad (39)$$
$$I = (V^*, ob, D, Y) \quad (40)$$

such that:

- V^* is the set of all strings constructed over the characters of V; in this formalism V^* represents the knowledge domain of I
- ob: $S \rightarrow V^*$ a injective function that maps the symbols of S into the strings of V^*
- $D = (V^*, \bullet)$ is a partial algebra over V^*. In what follows we will consider the operation $\bullet$ as a partial operation defined in the following manner:
  for every natural number m we take $\omega^{m+1} = \omega^m \bullet \omega$
- the set of algorithms $Y = \{Alg_u\}_{u \in L}$ generate the elements of the interpretation domain V^*, Alg$u$: (41).
-

$$V^* \times V^* \rightarrow V^*, u \in L \quad (41)$$

According to this interpretation system, the valuation mapping generated by I is defined as (42).

$$val_I : ASP(G) \rightarrow Y \quad (42)$$

such that (Dănciulescu, 2015):

- $val_I([x, y], u) = Alg_u(ob(x), ob(y)), u \in L_0$
- $val_I([x_1, \ldots, x_n], \sigma_L(u, v)) = val_I([x_1, \ldots, x_k], u) \bullet val_I([x_k, \ldots, x_n], v)$, for $\sigma_L(u, v) \in L$

Remark. (Dănciulescu, 2015) BecauseD = (V*, •) is a partial algebra, results that(43) if and only if: (44) and (45).

$$val_I([x_1, \ldots, x_k], u) \bullet val_I([x_k, \ldots, x_n], v) \in V^* \quad (43)$$
$$val_I([x_1, \ldots, x_k], u), val_I([x_k, \ldots, x_n], v) \in V^* \quad (44)$$
$$val_I([x_1, \ldots, x_k], u), val_I([x_k, \ldots, x_n], v) \in dom(\bullet) \quad (45).$$

Remark (Dănciulescu, 2015) For each (46), the string (47), is an element ofV*, therefore (48) represents the formal language overVconstructed by means of the abstract notations ofG.

$$(ob_1, ob_2) \in Y \times Y \quad (46)$$
$$Alg_u(ob_1, ob_2) \; for u \in L \quad (47)$$
$$\bigcup_{u \in L} Alg_u(ob_1, ob_2) \quad (48)$$

Definition (Dănciulescu, 2015) We define a Stratified Graph Representation System for formal language generation as follows:

$$SGRS = (G, (V^*, \bullet), ob, Y)$$

The inference process IP_{SGRS} generated by the system of knowledge representation *SGRS* is (49) (Dănciulescu, 2015).

$$IP_{SGRS}: ASP(G) \to Y \quad (49)$$

We obtain that the set *Y* that includes all the words constructed over the alphabet *V* by means of the algorithms defined in the interpretation system *I* by considering the representations of the stratified graph *G*.

Therefore (50) is an accepted structured path of the structure *G*, we have (51) (Dănciulescu, 2015).

$$\forall d \in ASP(G) \quad (50)$$

$$IP_{SGRS}(d) = val_I(d) \quad (51)$$

Following the inference process given above, the output elements of *Y* are defined as follows (Dănciulescu, 2015):

- if we take (52) results (53).

$$C_{(x,\, y)} = \{d \in ASP(G) \mid first(d) = x,\ last(d) = y\} \quad (52)$$

$$IP_{SGRS} = \bigcup_{(x,y) \in S \times S} IP_{SGRS}(C_{(x,y)}) \quad (53)$$

where: for (54) with (55) and (56).

$$IP_{SGRS} = \bigcup_{(x,y) \in S \times S} \{w \in Y \mid \exists d \in C_{(x,y)}: IP_{SGRS}(d) = w\} \quad (54)$$

$$d \in ASP(G) \quad (55)$$

$$d = ([x_1, \ldots, x_n], u),\ first(d) = x_1 \quad (56)$$

$$last(d) = x_n \quad (57)$$

3. Right-linear languages generated in systems of knowledge representation based on stratified graphs

A *Right Linear Grammar* is a *Context Free GrammarG* = (N, Σ, S, P) where each rule has one of the following forms: (58) and (59).

$$S_1 \to \alpha S_2 \text{ or } S_1 \to \alpha \text{ for } \alpha \in \Sigma^* \cup \{\lambda\} \quad (58)$$

$$S_1, S_2 \in N \quad (59)$$

A grammar is considered as a *Left Linear Grammar* if all the productions are of the form: (60) and (61).

$$S_1 \to S_2\alpha \text{ or } S_2 \to \alpha, \text{ for } \alpha \in \Sigma^* \cup \{\lambda\} \quad (60)$$

A *Regular Grammar* is a grammar that is either a Right Linear or a Left Linear Grammar.

In literature there are several generation mechanisms for Regular Languages like grammars, automata, transition networks and so on. In this study, we will consider the case of Right-Linear Languages generation; we will prove that we can emulate a Right-Linear Grammar using a system of knowledge representation based on stratified graphs by considering appropriate interpretations.

Indeed, let us consider *V* - a vocabulary. We can define a new mechanism for V^* words generation in a System of Knowledge Representation based on *SG* if we take (61).

$$SGRS = (\mathcal{G}, (V^*, \bullet), \{Alg_u\}_{u\in L}) \quad (61)$$

For $\mathcal{G} = (G_0, L, T, \sigma_T, f)$ a Stratified Graph over the labeled graph (62) for (63) - a partial algebra over the symbols of the vocabulary V and by $\bullet$ we note the string concatenation operation defined in above.

$$G_0 = (S, L_0, T_0, f_0) \quad (62)$$
$$(V^*, \bullet) \quad (63)$$

Based on the outputs given by the set of algorithms (64) we defined the *interpretationmapping* for the elements of $STR(G_0)$ as the function (65) such that:

- if $(x, y) \in f(a)$ then (66)
- if $(x, y) \in f(\sigma_L(u, v))$ then (67).

$$\{Alg_u\}_{u\in L} \quad (64)$$
$$val: STR(G_0) \rightarrow V^* \quad (65)$$
$$val(Path_a(x, y), a) = Alg_a(x, y) \quad (66)$$
$$val(Path_{\sigma_L(u,v)}(x, y), trace(\sigma_L(u, v))) = Alg_{\sigma_L(u,v)}(\omega_1, \omega_2) \quad (67)$$

Theorem. If (68) is a Right Linear Grammar, there exists a System of Knowledge Representation based on LSG such that (69).

$$G = (N, \Sigma_0, S_0, P) \quad (68)$$
$$L(G) = L(SGRS) \quad (69)$$

Proof. Let us considered (70) labeled graph such that:

- (71) is the set of nodes, each non-terminal symbol of the grammar G having a correspondent node in the graph G_0. In addition, for G_0 – the set of nodes we consider a special node noted with F that will be considered as the "final node".
- the set of arc labels of G_0 is the set of terminal symbols of the grammar G.
- for every production (72) there are (73) such that (74).
- for every production (75) we take (76).

$$G_0 = (X, \Sigma_0, T_0, f_0) \quad (70)$$
$$X = N \cup \{F\} \quad (71)$$
$$(S_1 \rightarrow \alpha S_2) \in P \quad (72)$$
$$S_1, S_2 \in X \quad (73)$$
$$(S_1, S_2) \in f_0(\alpha) \quad (74)$$
$$(S_1 \rightarrow \alpha) \in P \quad (75)$$
$$(S_1, F) \in f_0(\alpha) \quad (76)$$

We take (77) a stratified graph structure over the graph G_0. We will prove that for the Stratified Graph Representation System SGRS, defined based on the structure $\mathcal{G}$: (78).

$$\mathcal{G} = (G_0, L, T, f) \quad (77)$$
$$SGRS = (\mathcal{G}, (\Sigma_0, \bullet), \{Alg_u\}_{u\in L}) \quad (78)$$

We have L(SGRS) = $L(G)$. Indeed, let us consider the following n step of derivation in the grammar G (79)

$$\alpha_1\alpha_2 \ldots \alpha_n S_n \rightarrow \alpha_1\alpha_2 \ldots \alpha_n \alpha_{n+1} S_{n+1} \quad (79)$$

This step is modeled in the stratified graph $\mathcal{G}$ given in Figure 1 as it will be proved in what follows:

If (80), results that there are the following productions in the grammar G: (81) and by the production of the grammar G we have (82).

$$\omega = \alpha_1\alpha_2 \ldots \alpha_n \in L(G) \tag{80}$$

$$\begin{aligned} &S_0 \to \alpha_1 S_1 \\ &S_1 \to \alpha_2 S_2 \\ &\ldots \\ &S_k \to \alpha_{k+1} S_{k+1} \\ &\ldots \end{aligned} \tag{81}$$

$$S_0 \Rightarrow \alpha_1 S_1 \Rightarrow \alpha_1\alpha_2 S_2 \Rightarrow \ldots \Rightarrow \alpha_1 \ldots \alpha_{k+1} S_{k+1} \Rightarrow \ldots \Rightarrow \alpha_1 \ldots \alpha_n \; (*) \tag{82}$$

In order to model these derivations in the stratified graph $\mathcal{G}$, each production of the grammar will be represented in the labeled graph G_0 by a direct arc of the form given in Figure 2.

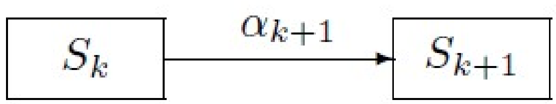

Figure 2. The representation of the rule $(S_k \to \alpha_{k+1} S_{k+1}) \in P$ in the labeled graph G_0

For each non-terminal symbol S_k of the grammar G there is a node S_k in the graph G_0 and for each production (83).

$$S_k \to \alpha_1 \ldots \alpha_{k+1} S_{k+1} \tag{83}$$

The labeled graph G_0 will have a relation (84), corresponding, in the stratified labeled graph $\mathcal{G}$, we will have (85).

$$(S_k, S_{k+1}) \in f_0(\alpha_{k+1}) \tag{84}$$

$$(S_0, S_{k+1}) \in f\,(\underbrace{\sigma_L(\sigma_L \ldots \sigma_L}_{k-times}(\alpha_1, \alpha_2), \ldots, \alpha_{k+1})) \tag{85}$$

Suppose (86), so that the Equation (*) is satisfied. We obtain (87) which implies the existence of (88) such that (89). This means that (90).

$$\omega = \alpha_1 \ldots \alpha_n \in L(G) \tag{86}$$

$$\begin{aligned} &(S_0, S_1) \in f_0(\alpha_1) \\ &\ldots \\ &(S_{n-1}, F) \in f_0(\alpha_n) \end{aligned} \tag{87}$$

$$d \in C_{(S_0, F)} \tag{88}$$

$$d = ([S_0, S_1, \ldots, S_n, F], \underbrace{\sigma_L(\sigma_L \ldots \sigma_L}_{(n-1) times}(\alpha_1, \alpha_2), \ldots, \alpha_n)) \tag{89}$$

$$\alpha_1 \ldots \alpha_n \in IP_{SGRS}(d) \subseteq L(SGRS) \tag{90}$$

Conversely, assume that (91). We have that (92), which implies that there is the sequence of non-terminals symbols (93) in the grammar G such that (94). We obtain that (95) and the theorem is proved.

$$\omega = \alpha_1 \ldots \alpha_n \in L(SGRS) \tag{91}$$

$$\exists d = ([S_0, S_1, \ldots, S_n, F], \underbrace{\sigma_L(\sigma_L \ldots \sigma_L}_{(n-1)\, times}(\alpha_1, \alpha_2), \ldots, \alpha_n)) \in C_{(S_0, F)} \quad (92)$$

$$S_0, \ldots, S_{n-1} \quad (93)$$

$$S_0 \Rightarrow \alpha_1 S_1 \Rightarrow \alpha_1 \alpha_2 S_2 \Rightarrow \ldots \Rightarrow \alpha_1 \ldots \alpha_n \quad (94)$$

$$\text{that } \omega \in L(G) \quad (95)$$

Example 1. Let us consider the following Right Linear Grammar (96), where (97). We obtain that (98) with a, $n \geq 0$.

$$G_1 = (\{S_0\}, \{a, b\}, S_0, P) \quad (96)$$

$$P = \{(S_0 \rightarrow abS_0), (S_0 \rightarrow a)\} \quad (97)$$

$$L(G_1) = (ab)^n \quad (98)$$

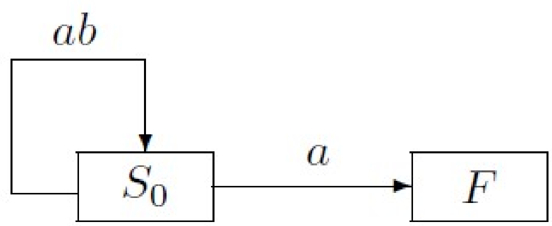

Figure 3. Representation of the grammar G_1 in the labelled graph G_0^1

If we take the labeled graph G_0^1 given in Figure 3 and construct the stratified graph structure over (99) such that (100) we obtain (101), (102).

$$G_0^1, G = (G_0^1, L, T, f) \quad (99)$$

$$L = \{ab, a, \sigma_L(ab, ab), \sigma_L(\sigma_L(ab, ab), a), \sigma_L(\sigma_L(ab, ab), ab), \sigma_L(\sigma_L(\sigma_L(ab, ab), ab), a), \ldots\} \quad (100)$$

$$(S_0, S_0) \in f(ab) \cup f(\sigma_L(ab, ab)) \cup f(\sigma_L(\sigma_L(ab, ab), ab)) \cup f(\underbrace{\sigma_L \ldots \sigma_L}_{n-times}(ab, ab), \ldots, ab)), \quad (101)$$

$$(S_0, F) \in f(a) \cup f(\sigma_L(ab, a)) \cup f(\sigma_L(\sigma_L(ab, ab), a)) \cup f(\underbrace{\sigma_L \ldots \sigma_L}_{n-times}(ab, ab), \ldots, a)) \quad (102)$$

Results that the language generated in the resulted system of knowledge representation over $\mathcal{G}$, (103) is (104). That is (105) with a, $n \geq 0$.

$$SGRS = (\mathcal{G}, (\{ab, a\}^*, \bullet), \{Alg_u\}_{u \in L}) \quad (103)$$

$$L(SGRS) = \bigcup_{d \in C_{(S_0, F)}} \{\omega \mid val(d) = \omega\} \quad (104)$$

$$L(SGRS) = (ab)^n \quad (105)$$

4. Conclusions and future work

In this paper, we proposed a new system for formal language generation by means of stratified graphs structures. This mechanism can generate languages of the first type and of the second type. More precisely, we propose a new system for formal language generation by means of a system of knowledge based on stratified graphs. We exemplified that, using an interpretation system specially defined for stratified graphs representations, a particular formal language can be obtained by means of the resulted accepted structured paths.

From our last work (Dănciulescu, 2015), several open problems still remain unsolved: the investigation of the manner in which, by imposing a set of restrictions in the inference process computations, the generated formal language sequences could be affected; the study the formal languages families that can be generated using this type of knowledge system: regular languages, context-sensitive languages, etc.

References

Dănciulescu, D. (2015). Formal Languages Generation in Systems of Knowledge Representation Based on Stratified Graphs. *Informatica,26*(3), 407-417. doi:10.15388/informatica.2015.55.

Dănciulescu, D., & Colhon, M. (2014). Splitting the structured paths in stratified graphs. Application in Natural Language Generation. *Analele Universitatii "Ovidius" Constanta - Seria Matematica,22*(2), 59-69. doi:10.2478/auom-2014-0031.

Dănciulescu, D. (2013) Systems Of Knowledge Representation Based On Stratified Graphs And Their Inference Process, In Proceedings of 9th International Conference of Applied Mathematics, Abstracts and Pre-Proceedings, Baia Mare.

Dănciulescu, D., Ţăndăreanu, N. (2013) Splitting the structured paths in stratified graphs, Modelling and Development of Intelligent Systems. In Proceedings of the 3th Int Conference on Modeling and Development of Intelligent Systems, University of Lucian Blaga from Sibiu Publishing House, ISSN 2067-3965.

Negru, V., Grigoraş, G., Dănciulescu, D. (2015) Natural Language Agreement in the Generation Mechanism based on Stratified Graphs. In Proceedings of the 7th Balkan Conference in Informatics (BCI 2015), Craiova, Romania.

Tudor, I. (2016) (2016). Properties of stratified languages. *Annals of the University of Craiova,43*(2), Mathematics and Computer Science series, 218-230. ISSN: 1223-6934

Tudor, (Preda), I.-V. (2010) Labelled Stratified Graphs can Generate Formal Languages. *European Conference for the Applied Mathematics and Informatics*, ISBN 978-960-474-260-8, 184-189.

Tudor (Preda), I.-V. (2011). Considerations Regarding Formal Languages Generation Using Labelled Stratified Graphs. *International Journal of Applied Mathematics and Informatics*, University Press, 2(5), 101-108.

Tudor (Preda), I.-V. (2011). Generating Natural Language by Labelled Stratified Graphs. *Scientific Bulletin*, University of Piteşti, Matematics and Informatics Series, 17, 89-100.

Ţăndăreanu, N. (2000). Knowledge bases with Output. *Knowledge ans Information Systems*, 2(4), 438-460.

Ţăndăreanu, N. (2003). Collaborations between distinguished representatives for labelled stratified graphs. *Annals of the University of Craiova*, Mathematics and Computer Science Series, 30(2), 184-192.

Ţăndăreanu, N. (2004). Knowledge representation by labelled stratified graphs. In Proceedings of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics, 5, 345-350.

Ţăndăreanu, N. (2005). *Labelled Stratified Graphs and their Applications in Knowledge Representation*, Technical Report, Research Centre for Artificial Intelligence, University of Craiova.

Ţăndăreanu, N. (2006). *Lecture Notes on Universal Algebra, Basic Concepts of Peano Algebras and Lattices*, Research Report in Artificial Intelligence, Research Centre for Artificial Intelligence, University of Craiova, Universitaria Publishing House.