DOI: 10.2478/auom-2014-0031
An. Şt. Univ. Ovidius Constanţa

VERSITA
Vol. 22(2),2014, 57–67

Splitting the structured paths in stratified graphs. Application in Natural Language Generation

Dana Dănciulescu and Mihaela Colhon

Abstract

The concept of labeled stratified graph (LSG) introduces some method of knowledge representation. The inference process developed for this structures uses the paths of the stratified graphs, an order between the elementary arcs of a path and some results of universal algebras. The order is defined by considering a structured path instead of a regular path. The application described in this paper interprets the symbolic elements of a LSG with natural language constructions. In this manner we obtained a mechanism for generation coherent texts in a natural language (for this approach, Romanian). The generation method is based on labeled stratified graph representation and the inference mechanism is guided by the structured paths of these representations.

1 Introduction

The concept of stratified graph provides a method of knowledge representation. This concept was introduced in paper [16]. The resulting method uses concepts from graph theory redefined in the new framework and elements of universal algebra. Intuitively, a stratified graph is built over a labeled graph placing on top a subset of a Peano algebra generated by the label set of considered graph.

The concept of structured path over a labeled graph was introduced in [14]. In the same paper was defined the concept of accepted structured path over a

Key Words: Peano algebra, labeled graph, stratified graph, structured path, accepted structured path, inference process, natural language generation
2010 Mathematics Subject Classification: Primary 08A55; Secondary 68W30.
Received: January, 2014.
Accepted: March, 2014.

stratified graph. The inference process was built by means of a decomposition property of the accepted structured path, described in an intuitive manner in [14].

The inference process developed in a stratified graph is based on the decomposition of an accepted structured path into two accepted structured paths. The resulted two components are subpaths of the initial path upon which a decomposition process is iterated until result only atomic accepted paths. A subpath defines a continuous path which consists of different kinds of elementary arcs of the initial path. The order induced by the structure of the accepted path and some meaning attached to every elementary arc are used in order to perform the inference.

In [15] the authors proposed a method by means of which a knowledge piece given in a natural language (English) is transposed in a labeled graph in order to construct inferences based on the given representations. Starting from this method, in this paper we define a natural language generation mechanism based on labeled stratified graph representation and inference.

2 Basic concepts

By *a labeled graph* we understand a tuple $G = (S, L_0, T_0, f_0)$, where S is a finite set of nodes, L_0 is a set of elements named *labels*, T_0 is a set of binary relations on S and $f_0 : L_0 \longrightarrow T_0$ is a surjective function. We consider a symbol σ of arity 2 and take the sets defined recursively as follows:

$$\begin{cases} B_0 = L_0 \\ B_{n+1} = B_n \cup \{\sigma(x_1, x_2) \mid (x_1, x_2) \in B_n \times B_n\}, n \geq 0 \end{cases}$$

where L_0 is a finite set that does not contain the symbol σ. The set $\mathcal{B} = \bigcup_{n\geq 0} B_n$ is the Peano σ -algebra ([12]) generated by L_0. We can understand that $\sigma(x, y)$ is the word σxy over the alphabet $L_0 \cup \{\sigma\}$. Often this algebra is denoted by $\overline{L_0}$.

By $Initial(\overline{L_0})$ we denote a collection of subsets of B satisfying the following conditions: $M \in Initial(\overline{L_0})$ if

- $L_0 \subseteq M \subseteq B$
- if $\sigma(u, v) \in M$, $u \in \overline{L_0}$, $v \in \overline{L_0}$ then $u \in M$ and $v \in M$

We define the mapping $prod_S : dom(prod_S) \longrightarrow 2^{S\times S}$ as follows:

$$dom(prod_S) = \{(\rho_1, \rho_2) \in 2^{S\times S} \times 2^{S\times S} \mid \rho_1 \circ \rho_2 \neq \emptyset\}$$

$$prod_S(\rho_1, \rho_2) = \rho_1 \circ \rho_2$$

where $\circ$ is the usual operation between the binary relations:

$$\rho_1 \circ \rho_2 = \{(x, y) \in S \times S \mid \exists z \in S : (x, z) \in \rho_1, (z, y) \in \rho_2\}$$

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

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

where $u \prec prod_S$ means that $dom(u) \subseteq prod_S$ and $u(\rho_1, \rho_2) = prod_S(\rho_1, \rho_2)$ for $(\rho_1, \rho_2) \in dom(u)$.

If u is an element of $R(prod_S)$ then we denote by $Cl_u(T_0)$ the *closure* of T_0 in the partial algebra $(2^{S\times S}, \{u\})$. This is the smallest subset Q of $2^{S\times S}$ such that $T_0 \subseteq Q$ and Q is closed under u. It is known that this is the union $\bigcup_{n\geq 0} X_n$, where

$$\begin{cases} X_0 = T_0 \\ X_{n+1} = X_n \cup \{u(\rho_1, \rho_2) \mid (\rho_1, \rho_2) \in dom(u) \cap (X_n \times X_n)\}, n \geq 0 \end{cases}$$

If $L \in Initial(L_0)$ then the pair $(L, \{\sigma_L\})$, 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 algebra. This property is used to define the concept of stratified graph.

Consider a labeled graph $G_0 = (S, L_0, T_0, f_0)$. A *stratified graph* ([16]) $\mathcal{G}$ over G_0 is a tuple (G_0, L, T, u, f) where

- $L \in Initial(\overline{L_0})$
- $u \in R(prod_S)$ 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 $(x, y) \in dom(\sigma_L)$

The existence of this structure, as well as the uniqueness is proved in [16].

Proposition For every labeled graph $G_0 = (S, L_0, T_0, f_0)$ and every $u \in R(prod_S)$ there is just one stratified graph (G_0, L, T, u, f) over G_0.

3 Graph-based Representations in Natural Language Processing

There are two sides to natural language processing. On the one hand, work in natural language understanding is concerned with the mapping from some

surface representation of linguistic material expressed as speech or text to an underlying representation of the meaning while maping from some underlying representation of meaning into text or speech is the domain of natural language generation [7]. Both are equally large and important problems, but the literature contains much less work on natural language generation (NLG) than it does on natural language understanding (NLU).

Graph theory is a well-studied sub-discipline of mathematics, with a large body of results and a large number of efficient algorithms that operate on graphs [11]. Despite the various existing linguistic theories, which lead to different ways of viewing sentence structure and therefore syntactic analysis, most linguists today agree that at the heart of sentence structure are the relations among words ([9], [10]). These relations refer either to grammatical functions (subject, complement etc.) or to links which bind words into larger units like phrases or even sentences [3]. A natural way to capture and process the connections between entities is by means of graph-based representations. The dependency graphs are well-known graph-based representations in which the syntactic and semantic features of sentences are depicted. In order to create dependency graphs, a sentence is processed by a dependency parser, which is based on the theoretical foundations of dependency grammar.

In 1979, Shapiro generates sentences from a semantic network. The Finite State Automata (FSA) as well as the Recursive Transitional Networks (shortly, RTNs) are considered recognizers (acceptors) of sentences generated by grammars [13] but such graph-based representations could be also used in NLG.

As mechanism for natural language processing, a RTN needs a lexicon of the language and a set grammar rules to break sentences into internal representations. Such representation consists of a set of nodes (states) and a set of labeled arcs (transitions) which usually have four types of labels for the transitions:
- $CAT \ < syn\_cat >$: this token must belong to a syntactic category given by $< syn\_cat >$ (like Noun, Verb, Adjective)
- $WORD \ < word\_form >$: this token must match the exact form of the label (usually a word form, such as upon)
- $PUSH \ < initial\_state\_subnetwork >$: conditional jump to the named subnetwork
- $JUMP$: an unconditional jump

An entry is accepted by a RTN if a final state is reached after all the input tokens were consumed.

Starting from the classical RTN representations, in the following section we propose a mechanism for text generation based on labeled stratified graphs whose arcs are labeled with generation conditions and the nodes with arbitrary

symbols used to mark the paths in the graph.

4 Natural Language Generation in Labeled Stratified Graphs

Parsing and as well as Natural Language Generation require a lexicon - a file of words giving their syntactic categories and lexical features, together with the inflectional forms of irregularly inflected words. Base forms, also known as lemmas or ground form do not contain any morphological derivation of the word (such as gender, number, tense, and so on) opposite to word forms which are made from the word base by adding inflectional morphemes.

Remark [5] In a specific language, a word form is uniquely identified by its lemma and the corresponding morpho-syntactic information. The reciprocal is not true: to a word form can correspond more morpho-syntactic interpretations, which have to be disambiguated by the context.

As noted by competent linguists, Romanian language is morphologically rich and relatively flexible word order language [2]. The term Morphologically Rich Languages refers to languages in which substantial grammatical information, i.e., information concerning the arrangement of words into syntactic units or cues to syntactic relations, are expressed at word level [4]. Some relevant word morphological attributes with respect to Romanian expressed based on their Part Of Speech data are:
- Verb: mood, time, person, number, gender
- Noun : number, gender, type
- Adjective: number, gender, degree
- Pronoun: type, gender, number, case
- Determiner: number, gender, type

Observation 1. Prepositions, adverbs, numerals and conjunctions have no morphological data.

In the case of Romanian, agreement between the syntactic components of a text is mandatory. For this reason, a generation system for texts in Romanian must checks for compatibility of the generated text (subject-verb agreement, article-head number agreement, gender compatibility, word-order, etc.). For example, an adjective in Romanian usually follows the noun it modifies and fully agrees with it in terms of number, gender, case, and definiteness.

As with parsing, we can represent the grammar in a Labeled Stratified Graph (LSG) if we allow the LSG to become a transition network for which arc labels refer to word categories and word forms. In this representation, each grammar rule can be transposed in a sequence of labeled paths. The symbols of the set L_0 can denote terminals that have associated some categories from lexic (word forms in the considered language) or nonterminals that denote

syntactic categories and which are not a priori instantiated with an element from lexicon (will be instantiated during the inference process).

In the course of text generation, the agreement rules must indicate how the combination of the syntactic categories and values must be associated with the syntactic elements [8]. The global task of NLG is to map a given formal input onto a natural language output to achieve a given communicative goal in a specific context ([1], [6]). The generation process we propose allows obtaining a surface text from regular paths chains. The natural language constructions are obtained using an inference process based on binary relations composition by paying attention that the generated constructions are well formed. During the generation, the agreement rules must force some values of the syntactic categories associated to some elements in order to agree with other elements.

4.1 Text Generation with Accepted Structured Paths

We consider a labeled graph $G_0 = (S, L_0, T_0, f_0)$. A *regular path* over G_0 is a pair $([x_1, \ldots, x_{n+1}], [a_1, \ldots, a_n])$ such that $(x_i, x_{i+1}) \in f_0(a_i)$ for every $i \in \{1, \ldots, n\}$.

The concept of structured path introduces some order between the arcs taken into consideration for a regular path.

bf Definition We denote the set of structured paths by $STR(G_0)$ the smallest set satisfying the following conditions:

- For every $a \in L_0$ and $(x, y) \in f_0(a)$ we have $([x, y], a) \in STR(G_0)$.
- If $([x_1, \ldots, x_k], u) \in STR(G_0)$ and $([x_k, \ldots, x_n], v) \in STR(G_0)$ then

$$([x_1, \ldots, x_k, \ldots, x_n], [u, v]) \in STR(G_0)$$

We define $STR_2(G_0) = \{w \mid \exists(\alpha, w) \in STR(G_0)\}$. We have that $STR_2(G_0)$ represents the projection of the set $STR(G_0)$ on the second axis.

We define the mapping $* : STR_2(G_0) \times STR_2(G_0) \longrightarrow STR_2(G_0)$ as follows:

- $dom(*) = \{(\beta_1, \beta_2) \mid \exists \alpha_1, \alpha_2 : (\alpha_1, \beta_1) \in STR(G_0), (\alpha_2, \beta_2) \in STR(G_0),$
  $last(\alpha_1) = first(\alpha_2)\}$
- If $\beta_1, \beta_2 \in dom(*)$ then $\beta_1 * \beta_2 = [\beta_1, \beta_2]$

Remark The pair $(STR_2(G_0), *)$ becomes a partial algebra.

Proposition The mapping $h : (STR_2(G_0), *) \longrightarrow ((\overline{L_0})_\sigma, \sigma)$ defined by

$$h(p) = \begin{cases} p \text{ if } p \in L_0 \\ \\ \sigma(h(u), h(v)) \text{ if } p = [u, v], u \in STR_2(G_0), v \in STR_2(G_0) \end{cases}$$

is a morphism of partial algebras.

Definition We define the set $ASP(\mathcal{G})$ as follows: $([x_1, \ldots, x_{n+1}], c) \in ASP(\mathcal{G})$ if and only if $([x_1, \ldots, x_{n+1}], c) \in STR(G_0)$ and $h(c) \in L$.
An element of $ASP(\mathcal{G})$ is named **accepted structured path** over $\mathcal{G}$.

We consider a stratified graph $\mathcal{G} = (G_0, L, T, u, f)$ over $G_0 = (S, L_0, T_0, f_0)$. Let note by Y a set of objects which includes all the nodes of S, that is $Y \supseteq S$. We suppose that for each $u \in L$ we have an algorithm $Alg_u : Y \times Y \longrightarrow Y$. This means that is a partial mapping such that $dom(Alg_u) \subseteq Y \times Y$ and for every pair $(x, y) \in dom(Alg_u)$ given as input for Alg_u this algorithm gives as output some element of Y.

Let us suppose that we have a description of the syntax of a natural language. We are not interested here what is the method used for the description but we suppose that it contains at least two elements: non-terminals (denoting the syntactic classes of the text components) and terminals (i.e. word forms). To highlight the role of structured paths in a NLG system based on labeled graph representations, we consider the example presented in Figure 1. We relieved here two accepted structured paths:
- one of them is denoted by (1) and represents the structured path $([x_1, x_2, x_3, x_4]$, $[[CAT\ Noun, CAT\ Noun], CAT\ Adj])$;
- the other is denoted by (2) and represents the structured path $([x_1, x_2, x_3, x_4]$, $[CAT\ Noun, [CAT\ Noun, CAT\ Adj]])$.

In order to explain in an intuitive manner the inference process defined for NLG we assign an algorithm to every arc symbol. For the example given in Figure 1, the following algorithms are attached to each labeled relation represented in the graph:

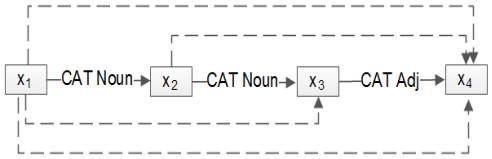

Figure 1: Intuitive representation of structured paths

Alg$_{CAT\ <syn\_cat>}$**(x,y)**
word_base_form ← pick a word from lexicon with the specified $< syn\_cat >$
Output: word_base_form
end

Alg$_{WORD\ <word\_form>}$**(x,y)**
Output: word_form
end

Alg$_{\sigma(CAT\ <syn\_cat1>,CAT\ <syn\_cat2>)}(o_1,\ o_2)$
IF $< syn\_cat2 >$ is a $Modifier\_Syntactic\_Class$ of $< syn\_cat1 >$
THEN
$o_2 \leftarrow$ word form of o_2: $o_2.number = o_1.number$, $o_2.gender = o_1.gender$
$o_2.case = o_1.case$
ENDIF
IF $< syn\_cat1 >=< syn\_cat2 >= Noun$ THEN
$o_2 \leftarrow$ word form of o_2: $o_2.case = oblique$
ENDIF
$this.head \leftarrow o_1$
Output: o_1 + " " + o_2
end

Alg$_{\sigma(u,v)}(o_1,\ o_2)_{u,v\in A\setminus A_0}$
IF $o_1.head$ and $o_2.head$ are $CAT\ Noun$ objects* THEN
$o_2 \leftarrow$ generate word form of o_2 for $o_2.case = oblique$
ENDIF
Output: o_1 + " " + o_2
end

In what follows all Romanian constructions are marked in quotes and the English equivalents follow the Romanian examples in brackets. Let us considet that $Alg_{CAT\ Noun}(x_1, x_2)$ = "băiatul" ("the boy"), $Alg_{CAT\ Noun}\ (x_2, x_3)$ = "mama" ("the mother"), $Alg_{CAT\ Adj}(x_2, x_3)$ = "frumos" ("beautiful"). Following the algorithms given above, we have:
$Alg_{\sigma(CAT\ Noun,CAT\ Noun)}(o_1, o_2)$ = "băiatul mamei" ("the mother's boy") with $o_1 = Alg_{CAT\ Noun}\ (x_1, x_2)$ = "băiatul" ("the boy"), $o_2 = Alg_{CAT\ Noun}$ (x_2, x_3) = "mama" ("the mother"). In "băiatul mamei" ("the mother's boy") the second word, represented by o_2, has oblique form. The morphosyntactic words features are:

"băiatul": number=sg., gender=masc., case=direct,

"mamei": number=sg., gender=fem., case=oblique.

$Alg_{\sigma(CAT\ Noun,CAT\ Adj)}(o_1, o_2)$ = "mama frumoasă" ("the beautiful mother") with $o_1 = Alg_{CAT\ Noun}\ (x_2, x_3)$ = "mama" ("the mother"), $o_2 = Alg_{CAT\ Adj}$ (x_3, x_4) = "frumos" ("beautiful"). The sequence "mama frumoasă" ("the beautiful mother") resulted from agreement in gender realization between o_1

*$CAT\ Noun$ objects are generated by $Alg_{CAT\ Noun}$ algorithms.

and o_2. The morphosyntactic words features are:

"mama": number=sg., gender=fem., case=direct,

"frumosă": number=sg., gender=fem.

$Alg_{\sigma(CAT\ Noun,\sigma(CAT\ Noun,CAT\ Adj))}(o_1, o_2)$= "băiatul mamei frumoase" ("the boy of the beautiful mother") with o_1 = "băiatul" ("the boy") , o_2 = "mama frumoasă" ("the beautiful mother"). In "băiatul mamei frumoase" ("the boy of the beautiful mother") the sequence of o_2 takes the oblique case. The morphosyntactic words features are:

"băiatul": number=sg., gender=masc., case=direct,

"mamei": number=sg., gender=fem., case=oblique,

"frumoase": number=sg., gender=fem., case=oblique.

$Alg_{\sigma(\sigma(CATNoun,CATNoun),CATAdj)}(o_1, o_2)$= "băiatul mamei frumos" ("the mother's beautiful boy") with o_1 = "băiatul mamei" ("the mother's boy"), o_2 = "frumos" ("beautiful"). In "băiatul mamei frumos" ("the mother's beautiful boy") the sequence of o_2 has the same gender, number and case with the head word of o_1, that is with "băiatul". The morphosyntactic words features are:

"băiatul": number=sg., gender=masc., case=direct,

"mamei": number=sg., gender=fem., case=oblique,

"frumos": number=sg., gender=masc., case=direct.

5 Conclusions and Future Works

In this paper we treat from the mathematical point of view the inference process based on stratified graphs by means of new concepts such as regular paths, structured paths and accepted structured paths. Also we proposed a new mechanism for interpreting the relations encoded in stratified graphs. This interpretation, defined in order to be used for natural language texts generation, pays attention to the agreement conditions that have to be fulfilled between the text components.

References

[1] Bouayad-Agha, N.; Casamayor, G., Wanner, L. *Natural language generation and semantic web technologies. Semantic Web Journal* (2012).

[2] Ceauşu, A.; Tufiş, D. *Addressing SMT Data Sparseness when Translating into Morphologically-Rich Languages*, Proceedings of NLPSC 2011, Special Issue Human-Machine Interaction in Translation, August 2011, Copenhagen, Denmark (2011).

[3] Colhon, M. *eRoL: Automatic Voice Translator for Romanian. Building Resources for a Symbolic Machine Translation Program*, Universitaria Publishing House, Craiova (2013).

[4] Colhon, M. *Acquiring Syntactic Translation rules from a Parallel Treebank*, Journal of Information and Library Science INFOtheca, **XIII(2)** (2012), 19-32.

[5] Colhon, M. Ţăndăreanu, N. *A Semantic Schema - based Approach for Natural Language Translation*, WSEAS Journal Transactions on Computers, **9(11)** (2010), 1307-1317.

[6] Dai, Y.; Zhang, S.; Chen, J.; Chen, T.; Zhang, W. *Semantic Network Language Generation based on a Semantic Networks Serialization Grammar*, World Wide Web **13(3)** (2010), 307-341.

[7] Dale, R.; Di Eugenio, B.; Scow, D. *Introduction to the special issue on natural language generation*, Computational Linguistics **24** (1998), 345-353.

[8] Diaconescu, Ş. *Natural Language Agreement Description for Reversible Grammars*, Advances in Artificial Intelligence Lecture Notes in Computer Science **2903** (2003), 161-172.

[9] Hristea, F.; Colhon, M. *Feeding Syntactic Versus Semantic Knowledge to a Knowledge-lean Unsupervised Word Sense Disambiguation Algorithm with an Underlying Naive Bayes Model*, Fundamenta Informaticae Journal **119(1)** (2012), 61-86.

[10] Hristea, F. *The Naive Bayes Model for Unsupervised Word Sense Disambiguation. Aspects Concerning Feature Selection*, Springer (2012).

[11] Mihalcea, R.; Radev, D. *Graph-Based Natural Language Processing and Information Retrieval*, ACL Association for Computational Linguistics **38(1)** (2012).

[12] Boicescu, V.; Filipoiu, A.; Georgescu G.;Rudeanu, S. *Lukasiewicz-Moisil Algebra*, Annals of Discrete Mathematics, **49** (1991).

[13] Simionescu, R. *Romanian Deep Noun Phrase Chunking using Graphical Grammar Studio*, Proceedings of the 8th International Conference ¨Linguistic Resources and Tools for Processing of the Romanian Language" (2011), 109-118.

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

[15] Ţăndăreanu, N.; Ghindeanu, M. *Image Synthesis from Natural Language Description*, Proceedings of 3rd Romanian Conference on Artificial Intelligence and Digital Communications, Craiova, Romania, **103** (2003), 5-15.

[16] Ţăndăreanu, N. *Proving the Existence of Labelled Stratified Graphs*, Annals of the University of Craiova **XXVII** (2000), 81-92

Dana DĂNCIULESCU,
Computer Science Department,
University of Craiova
Romania, 200585 Craiova, Alexandru Ioan Cuza, 13
Email: danadanciulescu@gmail.com

Mihaela Colhon,
Computer Science Department
University of Craiova
Romania, 200585 Craiova, Alexandru Ioan Cuza, 13
Email: mcolhon@inf.ucv.ro