(IN)DEPENDENCIES IN FUNCTIONAL GENERATIVE DESCRIPTION BY RESTARTING AUTOMATA

Martin Plátek, František Mráz, and Markéta Lopatková

Charles University in Prague, Czech Republic

Email: martin.platek@mff.cuni.cz, frantisek.mraz@mff.cuni.cz, lopatkova@ufal.mff.cuni.cz

Abstract

We provide a formal model of a stratificational dependency approach to natural language description. This formal model is motivated by an elementary method of analysis by reduction, which serves for describing correct sentence analysis. The model is based on enhanced restarting automata that assign a set of parallel dependency structures to every reduction of an input sentence. These structures capture the correspondence of dependency trees on different layers of linguistic description, namely the layer of surface syntax and the layer of language meaning. The novelty of this contribution consists in the formal extension of restarting automata in order to produce tree structures with several interlinked layers and in the application of these automata to the stratificational description of a natural language.

1. Introduction

We present a formal model of a stratificational dependency approach to natural language description. The proposed model is based on an elementary method of analysis by reduction (AR, see [8]). The analysis by reduction is modeled by the so-called enhanced restarting automata that assign a set of dependency structures (DR-structures) to every reduction of an input sentence. DR-structures can capture a set of dependency trees representing sentence on different layers of linguistic description in a parallel way. The novelty of this approach consists in the formal presentation of the stepwise parallel composition of tree structures on different language layers.

In [8], natural language description is modeled as a formal string to string translation using a suitable model of restarting automata. [11] introduces a class of enhanced restarting automata with an output consisting of a single DR-tree. Here we formally discuss a model able to represent several parallel dependency structures and thus to capture relations between syntactic structures on different layers derived from AR (see also [6]).

1.1. Functional Generative Description

The theoretical linguistic basis for our research is provided by the Functional Generative Description (FGD in the sequel), see esp. [12]. FGD is characterized by its stratificational and dependency-based approach to the language description.

The *stratificational approaches* split language description into layers, each layer providing complete description of a (disambiguated) sentence and having its own vocabulary and syntax. We use the version of FGD that distinguishes four layers of description:[1]

***t*-layer** (tectogrammatical layer) capturing deep syntax, which comprises language meaning in a form of a dependency tree; the core concepts of this layer are dependency, valency, and topic-focus articulation, see esp. [12];

***a*-layer** (analytical layer) capturing surface syntax in a form of a dependency tree (non-projective in general);

***m*-layer** (morphological layer) capturing morphology (string of triples [word form, lemma, tag]);

***w*-layer** (word layer) capturing individual words and punctuation marks in a form of a simple string.

There are one-to-one correspondences between individual symbols of the *w*- and *m*-layer (we leave aside small exceptions here) and between individual symbols of *m*- and *a*-layer; individual symbols of these three layers (surface layers in the sequel) reflect individual 'surface' words and punctuation marks. On the other hand, individual symbols of *t*-layer reflect only lexical words (the so-called function words, e.g. prepositions and auxiliary verbs, are captured as attributes of lexical words); moreover, surface ellipses (e.g. elided subject in Czech) are restored as nodes on *t*-layer.

Similarly as in other stratificational approaches (e.g. [9]), the layers are ordered; the lowest one being the simplest *w*-layer, the highest being the most abstract *t*-layer.

FGD as a *dependency-based approach* captures both surface and deep syntactic information in a form of dependency trees (formally delete-rewrite structures, DR-structures, see [4]). Words (i.e. their *a*- and *t*-correlates, respectively) are represented as nodes of the respective trees, each node being a complex unit capturing the lexical, morphological and syntactic features; relations among words are represented by oriented edges. The dependency nature of these representations is important particularly for languages with relatively high freedom of word order; it also complies with the shift of focus to deep syntactic representation, for which dependency approach is commonly used.

[1]We adopt the notation of the Prague Dependency Treebank [3], a large corpus of Czech sentences, which uses FGD as its theoretical basis.

1.2. Basic Principles of Analysis by Reduction

Analysis by reduction is based on a stepwise simplification of an analyzed sentence. It defines possible sequences of reductions (deletions) in the sentence – each step of AR is represented by (i) deleting at least one word of the input sentence, or (ii) by replacing an (in general discontinuous) substring of a sentence by a smaller substring. Consequently, it is possible to derive formal dependency relations between individual sentence members based on the possible order(s) of reductions.

Using AR we analyze an input sentence (w-layer) enriched with the metalanguage information from the m-, a- and t-layer. Symbols on different layers representing a single word of an input sentence are processed simultaneously.

The principles of AR can be summed up in the following observations:

- The fact that a certain word (or a group of words) can be deleted implies that this word (or group of words) *depends in* AR on one of the words retained in the simplified sentence; the latter being called *governing word(s) in* AR.
- Two words (or groups of words) can be deleted in an arbitrary order if and only if they are *mutually independent in* AR.
- In order to ensure adequate modeling of natural language meaning (on t-layer), certain groups of words have to be deleted in a single step (e.g. valency frame evoking words and their (valency) complementations [7]); such words are said to constitute a *reduction component*. Even in such cases, it is usual to determine governing-dependent pairs on the layer of surface syntax (a-layer). In such a case, it is necessary to define special rules for particular language phenomena.

When simplifying an input sentence, it is necessary to apply certain elementary principles assuring adequate analysis:

- **principle of correctness:** a grammatically correct sentence must be after its simplification correct as well; this principle is applied on all layers of language description;
- **principle of completeness:** a complete sentence after its simplification must be complete with respect to its valency structure as well, i.e. each frame evoking word must be 'saturated' on the t-layer [7];
- **principle of shortening:** at least one 'surface' word (i.e. its correlates on w-, m- and a-layer) must be deleted in each step of AR;
- **principle of minimality:** each step of AR must be 'minimal': any potential reduction step concerning less symbols in the sentence would violate the principle of completeness on the t-layer or the principle of correctness on the w-layer.

These principles imply that in a single reduction step, either (*case 1*) item(s) representing a single free modification or (*case 2*) items representing valency complementations of a single frame evoking word together with their governing word are processed on the *t*-layer. The sentence is simplified until the so-called *core predicative structure* is reached (typically formed by sentence predicate and its valency complementations). Let us consider the following Czech sentence.

Sídlo	*dnes*	*mohla*	*mít*	*ve*	*státě*	*Texas.*

residence - today - could - have - in - state - Texas

Eng: She (= elided Sb) could reside in the state of Texas today .

In [6], the basic principles of AR for the previous sample Czech sentence are exemplified on several reduction steps; namely, the step by step simplification of the sentence is described as well as the incremental building of its DR-structure (*a*- and *t*-trees). Similar stepwise reduction of other sample sentence can be found in [11].

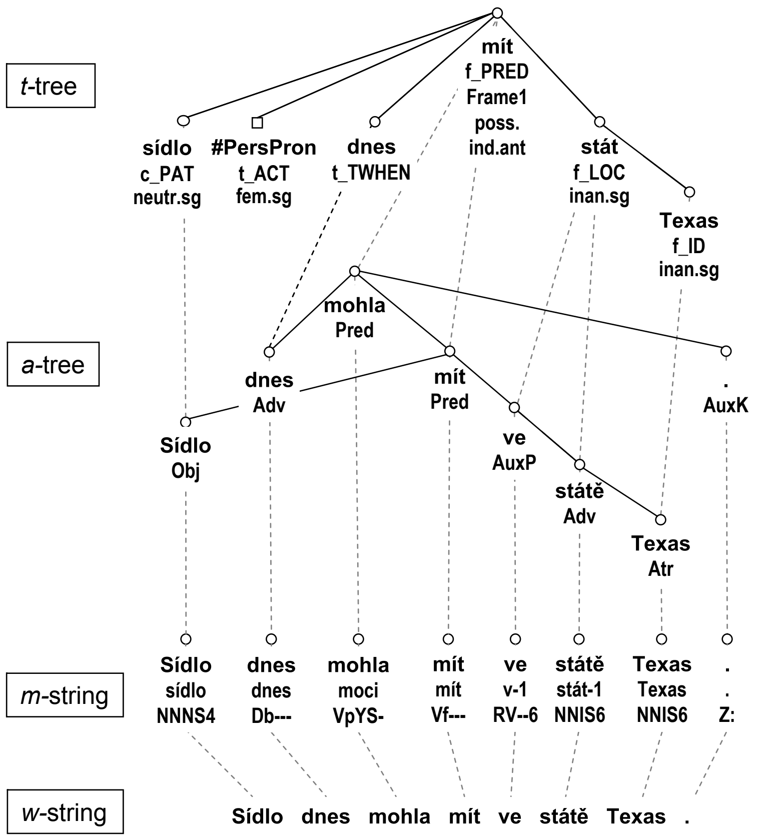


Figure 1: Parallel representation on *t*-, *a*-, *m*- and *w*-layers of the sample sentence according to FGD

Figure 1 shows the resulting structure corresponding to the previous sentence. It consists of the *t*-tree (deep syntactic tree representing the meaning), of the surface non-projective syntactic

a-tree, of the string of triples [word form, lemma, tag] on the m-layer and of the string of wordforms on w-layer. The dotted lines interconnect corresponding nodes. Let us focus on the non-trivial asymmetric relation between the a-layer and t-layer here: (i) the preposition *ve* 'in' as well as the noun *státé* 'state' in the a-tree are linked to the single t-symbol representing the lexical word *stát* 'state'; (ii) on the other hand, both the modal verb *mohla* 'could' and the lexical verb *mít* 'to have' are represented as the single t-node *mít* 'to have' (information on the modality is preserved as the attribute 'poss'); as a result, the non-projective a-tree is transformed to the projective t-tree; (iii) moreover, subject elided in the surface sentence is restored in the t-tree (the node with the symbol starting with #PersPron); thus this node has no counterpart on the a-layer.

Let us formulate basic requirements imposed on a DR-structure representing a natural language sentence:

- If a word w_1 depends on a word w_2 in a sentence u, then there is a path from w_1 to w_2 in the DR-structure D representing u.
- If words w_1 and w_2 are mutually independent in a sentence u, then they are not connected by a path in the corresponding DR-structure D.

In this paper, we focus mainly on the phenomena of valency, dependency and independency. We leave aside such phenomena as coordination, apposition and ellipsis in our model (such phenomena require a further enhancement of the model).

2. Restarting automata

First, we introduce a relevant type of a simple restarting automaton – sRL-automaton – rather informally. For technical reasons, we do it in a bit different way than in [10] and [11].

An sRL-*automaton* M is (in general) a nondeterministic machine with a finite-state control Q, a finite characteristic alphabet Γ, and a head (window of size 1) that works on a flexible tape delimited by the left sentinel ¢ and the right sentinel \$ ($¢, \$ \notin \Gamma$). For an input $w \in \Gamma^*$, the initial tape inscription is $¢w\$$. To process this input, M starts in its initial state q_0 with its window over the left end of the tape, scanning the left sentinel ¢. According to its transition relation, M performs the following operations in the individual steps:

- *moves to the right or to the left* – shifts the head one position to the right or left;
- $\mathrm{d_l}$ – deletes the visited symbol and shifts the head on its right neighbor;
- $\mathrm{w_r}[b]$ – rewrites the visited symbol by the symbol b;
- $\mathrm{p_b}$ – serves for marking (putting a numbered pebble on) the visited item only: marked items are used as nodes in DR-structures (in any other aspect it is an empty operation, see below);

- *accept* – halts the computations and accepts the input word.

Of course, neither the left sentinel ¢ nor the right sentinel \$ must be deleted. At the right end of the tape, M either halts and accepts, or it halts and rejects, or it *restarts*, that is, it places its window over the left end of the tape and reenters the initial state. It is required that prior to the first restart step and also between any two restart steps, M executes at least one delete operation. During each step, M can change its internal state.

We can see that any finite computation of M consists of certain phases. A phase, called a *cycle*, starts in a restarting configuration, the window is moved along the tape by performing its operations until a restart operation is performed and thus a new restarting configuration is reached. If no further restart operation is performed, each finite computation necessarily finishes in a halting configuration – such a phase is called a *tail.* We assume that no delete and rewrite operation is executed in a tail computation.

The notation $u \vdash^c_M v$ denotes a reduction performed during a cycle of M that begins with the tape inscription $¢u\$$ and ends with the tape inscription $¢v\$$; the relation $\vdash^{c^*}_M$ is the reflexive and transitive closure of $\vdash^c_M$.

A string $w \in \Gamma^*$ is *accepted* by M, if there is an accepting computation which starts in the restarting configuration with the tape inscription $¢w\$$ and ends by executing the *accept* operation. By $L_C(M)$ we denote the language consisting of all words accepted by M; we say that *M recognizes (accepts) the characteristic language $L_C(M)$.*

Remark 1. sRL-*automata are two-way nondeterministic automata, which allow to check the whole input sentence prior to any changes. It resembles a linguist who can read the whole sentence first, and reduce the sentence in a correct way afterwards. We choose a nondeterministic model to enable various orders of reductions. This can serve for the verification of the (in)dependency between the individual parts of a sentence.*

Similarly as in [10], we use a restricted type of sRL-automata for which the number of rewrite, delete and pebble operations made per cycle is limited by a constant. Such sRL-automata can be described by (meta)-instructions. The cycles of M are described by *restarting instructions* over Γ that describe the moves of the head and the changes of the states implicitly. Each cycle of M is described by a single restarting instruction over Γ of the following form:

$$I_{\mathrm{R}} = (¢ \cdot E_0, [a_1]_1 o_1, E_1, [a_2]_2 o_2, E_2, \ldots, E_{s-1}, [a_s]_s o_s, E_s \cdot \$, \mathsf{Restart}), \text{ where}$$

- $E_0, E_1, \ldots, E_s$ $(s > 0)$, are regular languages over Γ (usually represented by regular expressions);
- $o_1, \ldots, o_s \in \{\mathrm{d_l}, \mathrm{p_b}\} \cup \{\mathrm{w_r}[b] \mid b \in \Gamma\}$.
- The symbols $a_1, a_2, \ldots, a_s \in \Gamma$ are the symbols on which the corresponding operations $o_1, \ldots, o_s$ are executed.

For each operation o, let us define an auxiliary function $o : \Gamma \to \Gamma$ as:
$\mathrm{p_b}(a) = a, \quad \mathrm{d_l}(a) = \lambda, \quad \mathrm{w_r}[b](a) = b, \quad$ for all $a \in \Gamma$.

When trying to execute I_{R} starting from a tape inscription $¢w\$$, M will get stuck (and so reject), if w does not admit a factorization of the form $w = v_0 a_1 v_1 a_2 \ldots v_{s-1} a_s v_s$ such that $v_i \in E_i$ for all $i = 0, \ldots, s$. On the other hand, if w admits factorizations of this form, then one of them is chosen nondeterministically, and $¢w\$$ is transformed (reduced) into $¢v_0 o_1(a_1) v_1 \cdots v_{s-1} o_s(a_s) v_s\$$.

Tails of accepting computations are described by *accepting instructions* over Γ of the form $I_{\mathrm{A}} = (¢ \cdot E_0, [a_1]_1, E_1, [a_2]_2, E_2, \ldots, E_{s-1}, [a_s]_s, \; E_s \cdot \$, \mathsf{Accept})$, where individual E_i are finite languages over Γ. A tail performed by the instruction I_{A} starts with the inscription on the tape $¢z\$$; if $z \in E_0 a_1 \cdots a_s E_s$, then M accepts z (and the whole computation as well). Otherwise, the computation halts with rejection. This special form of accepting instruction is introduced with regard to the future enhancements of restarting automata.

Further we will refer to a sRL-automaton M as a tuple $M = (\Gamma, ¢, \$, R(M), A(M))$, where Γ is a characteristic vocabulary (alphabet), ¢ and \$ are sentinels not belonging to Γ, $R(M)$ is a finite set of restarting instructions over Γ, and $A(M)$ is a finite set of accepting instructions over Γ. The class of all sRL-automata will be denoted as sRL.

The following property of restarting automata has a crucial role in our applications of restarting automata.

(Correctness Preserving Property) A sRL-automaton M is *correctness preserving* if (for all $u, v \in \Gamma^*$)

$$u \in L_C(M) \text{ and } u \vdash_M^{c^*} v \text{ imply that } v \in L_C(M).$$

In order to model adequately the analysis by reduction, we only consider correctness preserving automata in the sequel.

2.1. DR-structures

In this subsection, we introduce DR-structures, which serve for the enhancement of the computations of restarting automata in the next section.

A DR-structure is a generalization of a DR-tree used in [4]. It is a directed acyclic graph $D = (V, H)$, where V is a finite set of nodes, and $H \subset V \times V$ is a finite set of edges. A *node* $u \in V$ is a tuple $u = [i, j, a]$, where a is a symbol assigned to the node, i, j are natural numbers, i represents the horizontal position of the node u, j represents the vertical position of u (it is equal to 0 or to the number of nodes with the same horizontal position i from which there are oriented paths to u). Each *edge* $h = (u, v) \in H$, where $u = [i_u, j_u, a]$ and $v = [i_v, j_v, b]$ for some nonnegative integers i_u, i_v, j_u, j_v and $a, b \in \Gamma$, is either

- *oblique edge* – if $i_u \neq i_v$, or

- *vertical edge* – if $i_u = i_v$ and $j_v = j_u + 1$.

We say that $D = (V, H)$ is a DR-tree if the graph D is a rooted inner tree (i.e. all maximal paths in T end in its single root).

2.2. Enhanced sRL-automata

In this section, we introduce enhanced restarting automata – the so-called sERL-automata. During their computations, these automata build DR-structures consisting of nodes containing deleted, rewritten, or marked symbols and of directed edges between them.

Enhanced sRL-*automata (*sERL-*automata)* were introduced in a restricted form in [11]. The model presented here works with more complex structures – in contrast to an sRL-automaton, an sERL-automaton can attach a DR-structure to any item of its tape. Each sERL-automaton M_e is actually an sRL-automaton with enhanced instructions.

An sERL-automaton $M_\mathrm{e} = (\Gamma, ¢, \$, ER(M_\mathrm{e}), EA(M_\mathrm{e}))$ consists of an alphabet Γ, sentinels $¢, \$ \notin \Gamma$, a set of enhanced restarting instructions $ER(M_\mathrm{e})$, and a set of enhanced accepting instructions $EA(M_\mathrm{e})$. An enhanced instruction is a pair $I_\mathrm{e} = (I, G)$ consisting of an instruction I of a sRL-automaton and a directed acyclic graph $G = (U, H)$ representing the required structure for symbols processed – deleted, rewritten or marked (pebbled) – during the application of the instruction I. The restrictions put on the set of edges of G are described below. Let us note that the graph G is described in a different way than the DR-structures; in this way, we stress their different purpose here.

Restarting instruction. If I is a restarting instruction $I = (¢ \cdot E_0, [a_1]_1 o_1, E_1, [a_2]_2 o_2, E_2, \dots, E_{s-1}, [a_s]_s o_s, E_s \cdot \$, \mathsf{Restart})$, then $o_1, \dots, o_s \in \{\mathrm{d_l}, \mathrm{p_b}\} \cup \{\mathrm{w_r}[b] \mid b \in \Gamma\}$ are the operations performed on symbols $a_1, \dots, a_s$. Let $o_{i_1} = \mathrm{w_r}[b_{i_1}], \dots, o_{i_r} = \mathrm{w_r}[b_{i_r}]$, for some $r \geq 0$, be all rewrite operations from $\{o_1, \dots, o_s\}$. The nodes in U are $1, 2, \dots, s, i'_1, i'_2, \dots, i'_r$ and they correspond to the symbols $a_1, \dots, a_s$ and symbols $b_{i_1}, \dots, b_{i_r}$, respectively. An edge $(u, v) \in H$ is:

1. either *deleting*: $u = i$ corresponds to a deleted symbol a_i (hence $o_i = \mathrm{d_l}$, $1 \leq i \leq s$) and $v = j$ for some j from U,
2. or *rewriting*: $u = i$ corresponds to a rewritten symbol a_i (hence $o_i = \mathrm{w_r}[b_{i_j}]$, $b_{i_j} \in \Gamma$, $1 \leq i \leq s$, $1 \leq j \leq r$) and $v = i'_j$.

Accepting instruction. If I is an accepting instruction $I = (¢ \cdot E_0, [a_1]_1 \mathrm{p_b}, E_1, [a_2]_2 \mathrm{p_b}, E_2, \dots, E_{s-1}, [a_s]_s \mathrm{p_b}, E_s \cdot \$, \mathsf{Accept})$, then the symbols $a_1, \dots, a_s$ are pebbled and they correspond to the nodes $\{1, 2, \dots, s\}$ of U. For each edge $h = (u, v) \in H$, $u \neq v$ and h is considered to be *deleting*.

2.3. Computations and (structured) languages by sERL-automata

Let $M_{\mathrm{e}} = (\Gamma, ¢, \$, ER(M_{\mathrm{e}}), EA(M_{\mathrm{e}}))$ be an sERL-automaton. By removing the graphs from the enhanced instructions M_{e}, we obtain instructions of the sRL-automaton $M = (\Gamma, ¢, \$, R(M), A(M))$, where $R(M) = \{I \mid (I, G) \in ER(M_{\mathrm{e}})\}$ and $A(M) = \{I \mid (I, G) \in EA(M_{\mathrm{e}})\}$. Then we define the reduction relation $\vdash_{M_{\mathrm{e}}}$ of M_{e} to be the same as the reduction relation $\vdash_M$ of M. That is, $u \vdash_{M_{\mathrm{e}}} v$ if M_{e} can execute a cycle starting with tape contents $¢u\$$ and finishing with $¢v\$$ on its tape. It implies that M_{e} accepts a word w iff M accepts w. Hence the characteristic languages of M_{e} and M coincide, $L_C(M_{\mathrm{e}}) = L_C(M)$.

A formalization of analysis by reduction is denoted as *characteristic reduction language*. The characteristic reduction language of the automaton M_{e} is the set $R_C(M_{\mathrm{e}}) = \{u \vdash_{M_{\mathrm{e}}} v \mid u, v \in L_C(M_{\mathrm{e}})\}$. This language fully determines enhanced computations of M_{e}, which are introduced in the next paragraphs. This property of enhanced sRL-automata illustrates the ability (and the method) of linguists to derive dependency structures from the analysis by reduction.

A (restarting) configuration $C = (T, D)$ of a given computation $\mathcal{C}$ by M_{e} consists of two parts: a set of items T representing current tape contents and a DR-structure D. The set T consists of a sequence of items of the form $[i, j, x]$, where i, j are nonnegative integers and $x \in \Gamma$. The value i (called horizontal position) determines original position of the respective input symbol in an input word. The value j (called vertical position) is the number of rewritings on the corresponding place of the input word since the start of the computation $\mathcal{C}$. This allows us to build a DR-structure carrying information on relations among symbols in an input word.

For a given input word $w = x_1 \dots x_n$ ($x_j \in \Gamma$, for $1 \le j \le n$), the initial contents of the tape is the set of items $T_w = \{[0, 0, ¢]\} \cup \{[i, 0, x_i] \mid i = 1, \dots, n\} \cup \{[n+1, 0, \$]\}$. During subsequent cycles, some symbols of the tape are deleted or rewritten. Rewritten items preserve the horizontal position of the items being rewritten, hence a tape inscription $¢x_1 \dots x_n\$$ ($x_i \in \Gamma$) can be represented by any set $T = \{[0, 0, ¢], [i_1, j_1, x_1], [i_2, j_2, x_2], \cdots, [i_n, j_n, a_n], [i_{n+1}, 0, \$]\}$, where $0 < i_1 < i_2 < \cdots < i_{n+1}$; such T is called a *(working) tape of* w. Obviously, the vertical position of both sentinels $¢$,$\$$ must be 0, and the horizontal position of the left sentinel must be zero, too. Let $Tp(w)$ denote the set of all tapes representing w.

The initial configuration $C_0 = (T_w, D_\emptyset)$ for an input word $w = x_1 \dots x_n$ consists of the set of items representing the initial contents of the tape T_w and the empty DR-structure $D_\emptyset = (\emptyset, \emptyset)$. By application of an enhanced restarting instruction I, a configuration C is transformed onto a configuration C'. The tape contents must be shortened and the DR-structure can grow.

In general, an *enhanced configuration of* M_{e} *on* w is a pair (P_w, D_w), where $P_w \in Tp(w)$, D_w is a DR-structure such that all maximal oriented paths in D_w end in nodes which belong to P_w. It is obvious that for any given word w and sERL-automaton M_{e}, there exists an infinite number of enhanced configurations of M_{e} on w.

Next, we will define an application of an enhanced restarting instruction. Let $I_r = (I, G)$

be an enhanced restarting instruction (see Section 2.2). An application of I_r in an enhanced configuration $C_w = (T, D)$ on a word w, where $T \in Tp(w)$, results in a new configuration $C' = (T', D')$. The application consists in the following steps:

1. Choosing a factorization of $\text{¢}w\$$ of the form $w = v_0 a_1 v_1 a_2 \dots v_{s-1} a_s v_s$ such that $v_i \in E_i$ for all $i = 0, \dots, s$. On the other hand, if w does not admit any factorization of this form, then I_{e} cannot be applied on C_w.
2. Rewriting the tape containing $\text{¢}w\$$ into the tape containing $\text{¢}v_0 o_1(a_1) v_1 \cdots v_{s-1}\ o_s(a_s) v_s \$$ in the following way:
   - each item from T that corresponds to an unchanged or pebbled symbols is copied onto T',
   - no item from T that corresponds to a deleted symbol is transferred onto T', and
   - each item $[i, j, x]$ from T that corresponds to a rewritten symbol $x = a_i$ (hence $o_i = \mathrm{w_r}[b_{i_j}]$, $b_{i_j} \in \Gamma$, $1 \le i \le s$, $1 \le j \le r$) is replaced in T' by the item $[i, j+1, b_{i_j}]$
3. For each edge $e \in H$ a new edge is inserted into the new DR-structure D'.

   If $e = (i, j)$ is a deleting edge, then an oblique edge is inserted leading from the item containing deleted a_i to the item corresponding to j (either a_j, if $1 \le j \le s$, or b_j, when $j \in \{i'_1, \dots, i'_r\}$) .

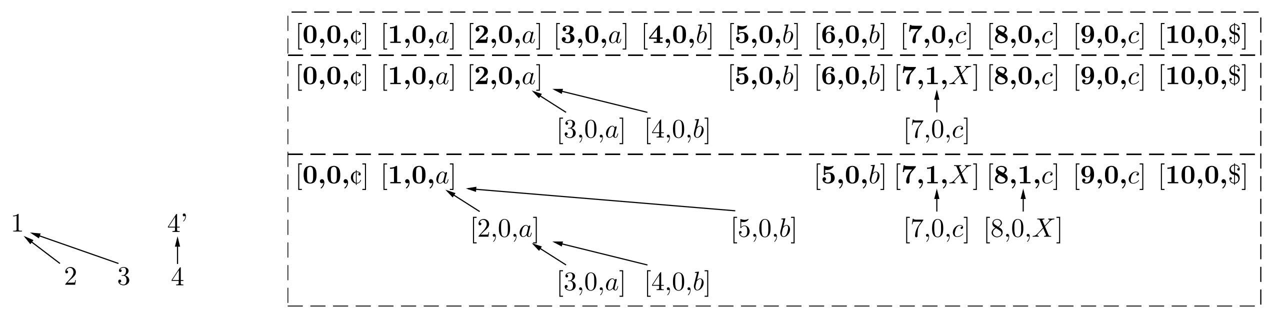
   If $e = (i, j)$ is a rewriting edge, then a vertical edge is inserted leading from the item containing a_i to the new item containing b_j ($j \in \{i'_1, \dots, i'_r\}$), which was inserted into T' in step 2.

   If there was a DR-structure attached to some of the deleted of rewritten cell, the structure is preserved and combined into a larger graph.

We say that the configuration C' can be reached in one *(restarting) step* from the configuration C_w by M_{e} (using the instruction I_{r}) and denote it by $C_w \models_{M_{\mathrm{e}}} C'$ (we also write $C_w \models^{I_{\mathrm{r}}}_{M_{\mathrm{e}}} C'$ if the instruction I_{r} ought to be specified).

Example 1. *Let* $I_1 = ((\text{¢} \cdot a^*, [a]_1\mathrm{p_b}, \lambda, [a]_2\mathrm{d_l}, \lambda, [b]_3\mathrm{d_l}, b^*X^*, [c]_4\mathrm{w_r}[X], c^* \cdot \$, \mathsf{Restart}), D_1)$ *be an enhanced restarting instruction with the graph* D_1 *(see the left part of Figure 2). Then two consecutive applications of* I_1 *on the word aaabbbccc will result in the following sequence of configurations (in the right part of Figure 2, the tape contents consist of the bold items depicted in the upper horizontal part of a picture for a particular configuration).*

Further, let us define an application of an enhanced accepting instruction $I_a = (I, G)$ in an enhanced configuration $C_w = (T, D)$ of M_{e} with a tape contents $T \in Tp(w)$ for some word w. The application of I_a in C_w results in a new configuration $C' = (T', D')$. Let $I = (\text{¢} \cdot E_0, [a_1]_1\mathrm{p_b}, E_1, [a_2]_2\mathrm{p_b}, E_2, \dots, E_{s-1}, [a_s]_s\mathrm{p_b}, E_s \cdot \$, \mathsf{Accept})$ and $G = (U, H)$. In this case, all the edges in H are interpreted as deleting edges. The application consists in:



1
2 3
4'
4
[0,0,¢] [1,0,a] [2,0,a] [3,0,a] [4,0,b] [5,0,b] [6,0,b] [7,0,c] [8,0,c] [9,0,c] [10,0,$]
[0,0,¢] [1,0,a] [2,0,a] [5,0,b] [6,0,b] [7,1,X] [8,0,c] [9,0,c] [10,0,$]
[3,0,a] [4,0,b] [7,0,c]
[0,0,¢] [1,0,a] [5,0,b] [7,1,X] [8,1,c] [9,0,c] [10,0,$]
[2,0,a] [5,0,b] [7,0,c] [8,0,X]
[3,0,a] [4,0,b]

Figure 2: The graph D_1 and the two steps on *aaabbbccc* using I_1

.

1. Choosing a factorization of $¢w\$$ of the form $w = v_0 a_1 v_1 a_2 \dots v_{s-1} a_s v_s$ such that $v_i \in E_i$ for all $i = 0, \dots, s$. On the other hand, if w does not admit any factorization of this form, then I_{a} cannot be applied on C_w.
2. Since only pebble operations are performed in I_{a}, the tape T is not changed ($T' = T$).
3. All edges and nodes from the DR-structure D are transferred into the new DR-structure D'. Moreover, for each edge $e = (i, j) \in H$ a new edge is inserted into D'. The new edge starts in the item representing a_i and ends in the item corresponding to a_j.

Also in this case, we say that the configuration C' can be reached in one *(accepting) step* from the configuration C by M_{e} using the instruction I_{a} and denote it by $C \models^{I_{\mathrm{a}}}_{M_{\mathrm{e}}} C'$.

Example 2. *Let* $I_2 = ((¢, [a]_1\mathrm{p_b}, \lambda, [b]_2\mathrm{p_b}, X^*, [c]_3\mathrm{p_b}, \$, \mathsf{Accept}), D_2)$ *be an enhanced accepting instruction with the graph* D_2 *from the left part of Figure 3. Then after applying* I_2 *on the resulting* DR*-structure from Example 1 we obtain the final* DR*-structure from the right part of Figure 3.*

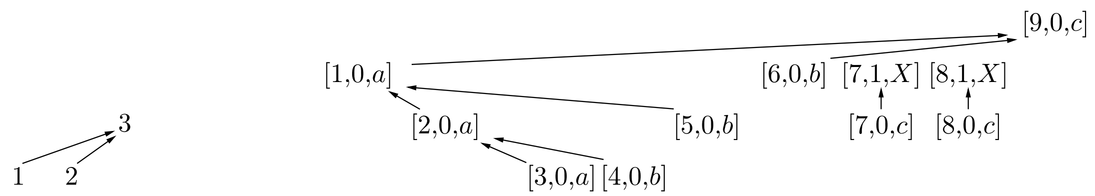


Figure 3: The graph D_2 and the final DR-structure on *aaabbbccc*.

Let $C \models_{M_{\mathrm{e}}} C'$, $C = (T, D)$, and $C' = (T', D')$. We denote as $\mathsf{IDR}(C, C')$ the smallest DR-structure which is a subgraph of D', and does not contain any edge from D. We say that $\mathsf{IDR}(C, C')$ is the DR-*increment* of the step $C \models_{M_{\mathrm{e}}} C'$.

Let $\mathcal{C} = C_0, C_1, \dots, C_k$ be a sequence of configurations such that $C_i \models^{I_i}_{M_{\mathrm{e}}} C_{i+1}$ holds for all $i = 0, \dots, k-1$, I_i are restarting instructions for all $i = 0, \dots, k-2$, I_{k-1} is an accepting instruction, and C_0 is the initial configuration for a word w. Then we say that $\mathcal{C}$ is an *accepting computation* of M_e, the word w is *accepted* by M_e, and the DR-structure D_k from the last configuration $C_k = (T_k, D_k)$ is the *output* DR-*structure* of the computation $\mathcal{C}$. We write $\mathsf{DR}(\mathcal{C}) = D_k$.

Let $AC(M_\mathrm{e})$ denote the set of all accepting computations of the sERL-automaton M_e. Then the DR-language of M_e is the set $\mathsf{DR}(M_\mathrm{e}) = \{\mathsf{DR}(\mathcal{C}) \mid \mathcal{C} \in AC(M_\mathrm{e})\}$.

2.4. Enhanced automata with several layers

First, we introduce a technical notion of *projection*. Let Σ and Γ ($\supset \Sigma$) be alphabets. The projection from Γ^* onto Σ^* denoted as Pr^Σ is the morphism defined as $a \mapsto a$ (for $a \in \Sigma$) and $A \mapsto \lambda$ (for $A \in \Gamma \smallsetminus \Sigma$). Similarly, we define the projection of languages: $\mathsf{Pr}^\Sigma : \mathcal{P}(\Gamma^*) \mapsto \mathcal{P}(\Sigma^*)$.

Similarly as above, we introduce a projection for DR-structures. Let D be a DR-structure from a DR-language over an alphabet Γ, $\mathsf{Pr}^\Sigma(D)$ denote a DR-structure over Σ that is obtained from D by removing all nodes with (at least one) symbol from $\Gamma \smallsetminus \Sigma$ and all edges incident to such nodes. In an obvious way, the projection can be extended into a projection of a DR-language over Γ.

Let $\Sigma_1, \ldots, \Sigma_j$, for some $j \geq 1$, be a sequence of pairwise disjoint alphabets and $\Gamma = \Sigma_1 \cup \cdots \cup \Sigma_j$. We say that sERL-automaton $M = (\Gamma, ¢, \$, ER(M), EA(M))$ is *an enhanced* sERL-*automaton with j layers* ((j)-sERL-automaton for short) if the following assumptions are fulfilled: M is *correctness preserving*, and M is allowed to rewrite a symbol from Σ_i (for $1 \leq i \leq j$) by a symbol from the same sub-vocabulary Σ_i, only. We refer to the symbols from Σ_i as the symbols on layer i (or i-symbols).

2.5. Current model of FGD as a (4)-sERL-automaton

Our ultimate goal is to model (in)dependencies in FGD – by our opinion, a (4)-sERL-automaton $\mathsf{M_{FD}}$ provides a suitable formal framework for the formal description of this part of the grammar of a natural language (Czech, in particular). In this way we formally separate (in)dependency from the more complex phenomena like coordination, apposition and ellipsis.

First, let us describe an important property of analysis by reduction modeled by $\mathsf{M_{FD}}$ (for all phenomena considered in FGD).

Reduction completeness: if $u, v \in L_C(\mathsf{M_{FD}})$, and v is a subsequence (a scattered substring) of u, then $u \vdash^*_{\mathsf{M_{FD}}} v$ holds.

Secondly, we describe the properties of $\mathsf{M_{FD}}$ that ensure the separation of the current version of $\mathsf{M_{FD}}$ from a more complex model capturing also phenomena other than (in)dependencies (like e.g. coordination):

- Let $C \models_{\mathsf{M_{FD}}} C_1$, $C \models_{\mathsf{M_{FD}}} C_2$, where C_1 and C_2 are different configurations. Then the

DR-increments $\mathsf{IDR}(C, C_1)$ and $\mathsf{IDR}(C, C_2)$ have disjoint sets of edges.

- We can see that the relation $\models_{\mathsf{M_{FD}}}$ (steps of $\mathsf{M_{FD}}$) creates a partial ordering on the set of configurations of $\mathsf{M_{FD}}$.

  Let us consider all (enhanced) computations starting from a starting configuration C_0; further, let us consider the sets of all steps St of $\mathsf{M_{FD}}$ and configurations C_f of these computations. We can see that the pair (C_f, St) (set and ordering) creates a semi-lattice, C_0 being its maximum. The second separation property for $\mathsf{M_{FD}}$ is formulated as a request on (C_f, St): the semi-lattice (C_f, St) has to be a lattice as well, i.e. it must have also a (common) single minimum.

  Let us stress that if we would extend $\mathsf{M_{FD}}$ for analysis of sentences with coordination or apposition, the corresponding semi-lattice will have at least two different minimal configurations.

- The DR-structures computed by all computations that start from a starting configuration C_0 are equal. Moreover, all these computations produce the same set of DR-increments (naturally in different sequences).

Further, we continue with the technical properties and notions connected with $\mathsf{M_{FD}}$.

Let $\mathsf{M_{FD}} = (\Gamma, ¢, \$, ER(\mathsf{M_{FD}}), EA(\mathsf{M_{FD}}))$; Γ consists of four parts $\Gamma = \Sigma_w \cup \Sigma_m \cup \Sigma_a \cup \Sigma_t$, which correspond to the respective layers of FGD (Section 1.2). Recall that the symbols from individual layers can be quite complex.

A *language of layer* $\ell \in \{w, m, s, t\}$ *accepted by* $\mathsf{M_{FD}}$ is obtained as a projection of the characteristic language onto Σ_ℓ, i.e. $L_\ell(\mathsf{M_{FD}}) = \mathsf{Pr}^{\Sigma_\ell}(L_C(\mathsf{M_{FD}}))$.

The characteristic language $L_C(\mathsf{M_{FD}})$ contains input sequences (over Σ_w) interleaved with (meta)language information in the form of symbols from $\Sigma_m \cup \Sigma_a \cup \Sigma_t$. Hence, the language of correct Czech sentences is $L_w = \mathsf{Pr}^{\Sigma_w}(L_C(\mathsf{M_{FD}}))$.

Similarly, a DR-*language of layer* ℓ *accepted by* $\mathsf{M_{FD}}$ is obtained as $\mathsf{DR}_\ell(\mathsf{M_{FD}}) = \mathsf{Pr}^{\Sigma_\ell}(\mathsf{DR}(\mathsf{M_{FD}}))$, $\ell \in \{w, m, s, t\}$. Let us note that $\mathsf{DR}_w(\mathsf{M_{FD}})$ and $\mathsf{DR}_m(\mathsf{M_{FD}})$ are empty (L_w and L_m are string languages). Further, $\mathsf{DR}_a(\mathsf{M_{FD}})$ and $\mathsf{DR}_t(\mathsf{M_{FD}})$ are languages of DR-trees. Each DR-tree $T \in \mathsf{DR}_t(\mathsf{M_{FD}})$ is *projective* (with respect to its descendants); that is, for each node n of the DR-tree T all its descendants constitute a contiguous sequence in the horizontal ordering of nodes of the tree T. On the other hand, trees from $\mathsf{DR}_a(\mathsf{M_{FD}})$ can be in general non-projective.

The DR-language $\mathsf{DR}_t(\mathsf{M_{FD}})$ represents the set of meaning descriptions in FGD whereas $\mathsf{DR}_a(\mathsf{M_{FD}})$ models a set of (surface) syntactic trees.

Let $z \in L_C(\mathsf{M_{FD}})$ and let $\mathsf{DR}(z, \mathsf{M_{FD}})$ denote the DR-structure from $\mathsf{DR}(\mathsf{M_{FD}})$ resulting from (all) accepting computations of $\mathsf{M_{FD}}$ on z. Let us recall that the language description based on

FGD is disambiguated (see Section 1.1); in particular, $\mathsf{DR}_t(\mathsf{M}_{\mathsf{FD}})$ and $\mathsf{DR}_a(\mathsf{M}_{\mathsf{FD}})$ contain exactly one t-tree and a-tree, respectively, resulting from all accepting computations of M_{FD} on z.

Let us note that $L_t(\mathsf{M}_{\mathsf{FD}})$ is designed as a deterministic context-free language. Readers familiar with restarting automata can see that $L_C(\mathsf{M}_{\mathsf{FD}})$ is a deterministic context-sensitive language and $L_w(\mathsf{M}_{\mathsf{FD}})$ is a context-sensitive language.

So far, we have not mentioned the edges from DR-structures from $\mathsf{DR}(\mathsf{M}_{\mathsf{FD}})$ that interconnect nodes (their symbols) in different layers. These edges serve for connecting the corresponding lexical units on different layers (see the dotted lines in Figure 1), they are obtained by applications of extended restarting meta-instructions of M_{FD}.

Here such an edge connects nodes on neighboring layers only. For M_{FD}, these edges can connect w-layer nodes to m-layer nodes, m-layer nodes to a-layer nodes, a-layer nodes to t-layer nodes, and nothing else (see Figure 1).

Let $z \in L_C(\mathsf{M}_{\mathsf{FD}})$ and let $\mathsf{DR}_t(z, \mathsf{M}_{\mathsf{FD}})$ denote the DR-tree from $\mathsf{DR}_t(\mathsf{M}_{\mathsf{FD}})$ resulting from any accepting computation of M_{FD} on z (see above the property of unambiguity). Now we can define the TR-*characteristic relation* $\mathsf{TSH}(\mathsf{M}_{\mathsf{FD}})$ of the automaton M_{FD}, which is the relation between sentence on the (w-layer) and its meaning (t-layer), as

$$\mathsf{TSH}(\mathsf{M}_{\mathsf{FD}}) = \{(u,t) \mid \exists y \in L_C(\mathsf{M}_{\mathsf{FD}}), u = \mathsf{Pr}^{\Sigma_w}(y) \text{ and } t \in \mathsf{DR}_t(y, \mathsf{M}_{\mathsf{FD}})\}.$$

Remark 2. TR-*characteristic relation represents the most important relations in FGD – the relations of synonymy and ambiguity. From the characteristic relation, the fundamental notions of analysis and synthesis are derived.*

A TSH-*synthesis* using M_{FD} is the function that returns the set of all word forms (i.e. Czech sentences) u which are in TSH relation with a given DR-tree $t \in \mathsf{DR}_t(\mathsf{M}_{\mathsf{FD}})$. Formally:

$$\textit{synt}\text{-}\mathsf{TSH}(\mathsf{M}_{\mathsf{FD}}, t) = \{u \mid (u,t) \in \mathsf{TSH}(\mathsf{M}_{\mathsf{FD}})\} \ .$$

For a tectogrammatical representation, i.e. a DR-tree t from $\mathsf{DR}(\mathsf{M}_{\mathsf{FD}})$, $\textit{synt}\text{-}\mathsf{TSH}(\mathsf{M}_{\mathsf{FD}}, t)$ is the set of all corresponding sentences from $L_w(\mathsf{M}_{\mathsf{FD}})$. Informally, $\textit{synt}\text{-}\mathsf{TSH}(\mathsf{M}_{\mathsf{FD}}, t)$ is the set of all sentences having 'meaning' t. This notion makes it possible to study synonymy of sentences (when a sentence has several meanings) and its degree.

A function dual to the TSH-synthesis is a function called TSH-*analysis of a string u using* M_{FD}:

$$\textit{anal}\text{-}\mathsf{TSH}(\mathsf{M}_{\mathsf{FD}}, u) = \{t \mid (u,t) \in \mathsf{TSH}(\mathsf{M}_{\mathsf{FD}})\} \ .$$

For a given sentence $u \in L_w$, TSH-analysis returns all its possible tectogrammatical representations from $\mathsf{DR}(\mathsf{M}_{\mathsf{FD}})$ – i.e. all 'meanings' of u.

Hence it models ambiguity of an individual Czech sentence u. This notion represents a formal definition of complete syntactic-semantic analysis using M_{FD}.

Similarly, we can define a surface syntactic relation and a surface analysis. Let $z \in L_C(\mathsf{M}_{\mathsf{FD}})$ and let $\mathsf{DR}_a(z, \mathsf{M}_{\mathsf{FD}})$ denote the (single) DR-tree from $\mathsf{DR}_a(\mathsf{M}_{\mathsf{FD}})$ resulting from accepting computations of M_{FD} on z.

A *surface syntactic relation* $\mathsf{TSX}(\mathsf{M}_{\mathsf{FD}})$ of the automaton M_{FD} is defined as the relation between a Czech sentence (on the w-layer) and its surface syntax (a-layer).

$$\mathsf{TSX}(\mathsf{M}_{\mathsf{FD}}) = \{(u,t) \mid \exists y \in L_C(\mathsf{M}_{\mathsf{FD}}), u = \mathsf{Pr}^{\Sigma_w}(y) \text{ and } t \in \mathsf{DR}_a(y, \mathsf{M}_{\mathsf{FD}})\}.$$

Finally we introduce the notion of TSX-*analysis of a string u using* M_{FD}:

$$anal\text{-}\mathsf{TSX}(\mathsf{M}_{\mathsf{FD}}, u) = \{t \mid (u,t) \in \mathsf{TSX}(\mathsf{M}_{\mathsf{FD}})\}\ .$$

TSX-analysis serves as the (surface) syntactic analysis for Czech in FGD. We will show elsewhere that, in general (and for Czech in particular), the TSX-analysis (surface syntactic relation) cannot adequately substitute the TSH-analysis (TR-characteristic relation), i.e. the surface syntactic relation cannot adequately describe the synonymy and ambiguity at the same time. On the other hand, the procedure of complete TSH-analysis is usually implemented in two basic steps, the first of them being the TSX-analysis. Similarly, the TSH-synthesis is usually implemented in several steps.

Concluding remarks

In this paper, encouraged by [2, 1], we extend the formal model of natural language description based on FGD so that it outputs the so-called reduction language and a set of complex DR-structures. In this way we give a formalization of the basic methodology of FGD and the basic task of capturing (in)dependencies in natural language sentences in terms of the automata theory. Let us stress that this model does not work with any other symbols than Czech word-forms and linguistic categories (meta-symbols); these categories can be found in grammar books describing the Czech language and in Czech lexicons. This means that the model does not use any auxiliary symbol(s) for technical reasons only. This issue is one of the important differences between current models and the models of FGD from the previous century.

The novelty of this contribution consists in a further formal extension of restarting automata in order to produce dependency based tree structures with several interlinked layers and in the application of these automata to the stratificational description of a natural language. Let us note that the new model can be considered also as an extension and generalization of Free-Order Dependency Grammars (see [4]). Therefore, it can be used for the study of the complexity and freedom of the word-order of natural languages by dependency structures, namely for the complexity and freedom of word-order of the Czech language. The measures defined in [4] can be used also for DR-structures produced by sERL-automata. We envisage that the proposed methodology is not FGD-specific. A similar formal frame can be prepared for other language descriptions, as e.g. for those presented in [9] and [5].

In the near future, we plan to study the relation between reduction languages and DR-languages in detail. We believe that we will be able to show more explicitly the adequateness of the presented model for FGD.

Acknowledgements

The paper reports on the research supported by the grants of GA ČR P202/10/1333, P103/10/0783, and 405/08/0681. It is carried under the project of MŠMT ČR MSM0021620838.

References

[1] BENSCH, S., DREWES, F., Millstream Systems, Report UMINF 09.21, Umeå University, 2009.

[2] GRAMATOVICI, R., MARTÍN-VIDE, C., Sorted Dependency Insertion Grammars, Theor. Comput. Sci. **354**(1) (2006) 142–152.

[3] HAJIČ, J. et al., Prague Dependency Treebank 2.0, Linguistic Data Consortium, Philadelphia, PA, USA 2006.

[4] HOLAN, T., KUBOŇ, V., OLIVA, K., PLÁTEK, M., Two Useful Measures of Word Order Complexity, in: A. Polguere, S. Kahane (Eds.) Processing of Dependency-Based Grammars. Workshop Proceedings (COLING/ACL'98), Montreal, ACL, 1998, 21–28.

[5] KUNZE, J., Abhängigkeitsgrammatik, Volume XII of Studia Grammatica, Akademie Verlag, Berlin 1975.

[6] LOPATKOVÁ, M., MRÁZ, F., PLÁTEK, M., Towards a Formal Model of Natural Language Description Based on Restarting Automata with Parallel DR-structures, in: Proceedings of ITAT, 2010 (accepted).

[7] LOPATKOVÁ, M., PLÁTEK, M., KUBOŇ, V., Modeling Syntax of Free Word-Order Languages: Dependency Analysis by Reduction, in: Matoušek, Mautner, Pavelka (Eds.) Proceedings of TSD 2005. LNAI 3658, Springer-Verlag, Berlin Heidelberg 2005, 140–147.

[8] LOPATKOVÁ, M., PLÁTEK, M., SGALL, P., Towards a Formal Model for Functional Generative Description, Analysis by Reduction and Restarting Automata, The Prague Bulletin of Mathematical Linguistics **87** (2007) 7–26.

[9] MEĽČUK, I. A., Dependency Syntax: Theory and Practice, State University of New York Press, Albany 1988.

[10] MESSERSCHMIDT, H., MRÁZ, F., OTTO, F., PLÁTEK, M., Correctness Preservation and Complexity of Simple RL-Automata, in: Implementation and Application of Automata. LNCS 4094, Springer-Verlag, Berlin Heidelberg 2006, 162–172.

[11] PLÁTEK, M., MRÁZ, F., LOPATKOVÁ, M., Restarting Automata with Structured Output and Functional Generative Description, in: A. Dediu, H. Fernau, C. Martin-Vide (Eds.) Proceedings of LATA 2010, LNCS 6031. Springer-Verlag, Berlin Heidelberg, 2010, 500–511.

[12] SGALL, P., HAJIČOVÁ, E., PANEVOVÁ, J., The Meaning of the Sentence in Its Semantic and Pragmatic Aspects, Reidel, Dordrecht 1986.