FDSL 6.5, 173-190

Markéta Lopatková, Martin Plátek, and Petr Sgall

Functional Generative Description, Restarting Automata and Analysis by Reduction[*]

1. Introduction

Functional Generative Description (FGD is a dependency based system for Czech, whose beginnings date back to the 1960s (see esp. Sgall *et al.* 1969, Sgall *et al.* 1986). FGD may be of some interest for the description of most Slavic languages, since it is adapted to treat a high degree of *free word order*. It not only specifies surface structures of the given sentences, but also translates them into their underlying representations. These representations (called tectogrammatical representations, denoted TRs) are intended as an appropriate input for a procedure of semantico-pragmatic interpretation in the sense of intensional semantics (see Hajičová *et al.* 1998). Since TRs are, at least in principle, disambiguated, it is possible to understand them as rendering linguistic (literal) meaning (whereas figurative meaning, specification of reference and other aspects belong to individual steps of the interpretation).

FGD has been implemented as a generative procedure by a sequential composition of pushdown automata (see Sgall *et al.* 1969, Plátek *et al.* 1978). Lately, as documented e.g. in Petkevič (1995), we have been interested in the formalization of FGD designed in a declarative way. In the present paper we want to formulate a formal framework for the procedure of checking the appropriateness and completeness of a description of a language in the context of FGD. The first step in this direction was introduced in Plátek (1982), where the formalization by a sequence of translation schemes is interpreted as an analytical system, and as a generative system as well. Moreover, requirements for a formal system describing a natural language *L* have been formulated – such a system should capture the following issues:

- The set of correct sentences of the language *L*, denoted by *LC*.
- The formal language *LM* representing all possible tectogrammatical representations (TRs) of sentences in *L*.
- The relation *SH* between *LC* and *LM* describing the ambiguity and the synonymy of *L*.

[*] This paper is a result of the project supported by the grants No. 1ET100300517 and MSM0021620838. The extended version is prepared for The Prague Bulletin of Mathematical Linguistics.

- The set of the correct structural descriptions *SD* representing in a structural way all possible TRs of sentences in *L* as dependency-based structures (*dependency trees*).

The object of the present paper concerns the foundations of a *reduction system* which is more complex than a reduction system for a (shallow) syntactic analyzer, since it provides not only the possibility of checking the well-formedness of the (surface) analysis of a sentence, but its underlying (tectogrammatical in terms of FGD) representation as well. Such a reduction system makes it possible to define formally the *analysis* as well as the *synthesis* of a sentence.

We propose here a new formal frame for checking FGD linguistic descriptions, based on *restarting automata*, see e.g. Otto (2006), Messerschmidt *et al.* (2006). We fully consider the first three requirements, i.e., *LC*, *LM* and *SH*. The fourth one is not formally treated here.

The main contribution of the new approach consists in the fact that it mirrors straightforwardly the so-called *(multi-level) analysis by reduction*, an implicit method used for linguistic research. Analysis by reduction consists of stepwise correct reductions of the sentence; roughly speaking, the input sentence is simplified until the so-called *core predicative structure* of the sentence is reached. It allows for obtaining (in)dependencies by the *correct reductions* of Czech sentences as well as for describing properly the complex word-order variants of a language with a high degree of `free' word order (see Lopatková *et al.* 2005). During the analysis by reduction, a (disambiguated) input string is processed, i.e., a string of tokens (word forms and punctuation marks) enriched with metalanguage categories from all linguistic layers encoded in the sentence.

In Section 2., we provides a brief characterization of analysis by reduction (subsection 2.1.) and then we address two basic linguistic phenomena, dependency (subsection 2.2.) and word order (2.3.), and show the process of the analysis by reduction on examples from Czech.

Now, let us briefly describe the type of restarting automaton that we use for modeling analysis by reduction for FGD (see Section 3). A *4-LRL-automaton* M_{FGD} is a non-deterministic machine with a finite-state control Q, a finite characteristic vocabulary Σ (see below), and a head (window of size 1) that works on a flexible tape. The automaton M_{FGD} performs:

- *move-right* and *move-left steps*, which change the state of M_{FGD} and shift the window one position to the right or to the left, respectively,
- *delete steps*, which delete the content of the window, thus shortening the tape, change the state, and shift the window to the right neighbor of the symbol deleted.

At the right end of the tape, M_{FGD} either halts and *accepts* the input sentence, 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 before the first restart step and also between any two restart steps, M_{FGD} executes at least one delete operation.

The *4-LRL*-automata can be also represented by a final set of so called metarules (see Messerschmidt *et al.* 2006), a declarative way of representation, which seems to be a very promising tool for natural language description.

The basic notion related to M_{FGD} is the notion of the language accepted by M_{FGD}, so called *characteristic language* $L_\Sigma(M_{FGD})$. In our approach, it is considered as a language that consists of all sentences from the surface language *LC* over alphabet Σ_0 enriched with metalanguage information from Σ_1, Σ_2, Σ_3. The tectogrammatical language *LM* as well as the relation *SH* can be extracted from $L_\Sigma(M_{FGD})$.

In order to model the analysis by reduction for FGD, the *4-LRL*-automaton M_{FGD} works with a complex characteristic vocabulary Σ that is composed from (sub)vocabularies Σ_0, … Σ_3. Each subvocabulary Σ_i represents the corresponding layer of language description in FGD, namely:

- Σ_0 is the set of Czech written *word-forms* and *punctuation marks* (tokens in the sequel), it is the vocabulary for the language *LC* from the request 1 above;
- Σ_1 represents the *morphemic layer* of FGD, namely morphological lemma and tag
- for each token;
- Σ_2 describes surface syntactic functions (as e.g., Subject, Object, Predicate);[1]
- Σ_3 is the vocabulary of the *tectogrammatical layer* of FGD describing esp. `deep' roles, valency frame for frame evoking words, and meaning of morphological categories.

That means that the automaton has an access to all the information encoded in the processed sentence (as well as a human reader/linguist has all the information for his/her analysis).

M_{FGD} was introduced with no ambitions to model directly the procedure of the sentence-generating in the human mind or of the procedure of understanding performed in the human mind. On the other hand, it has a straightforward ambition to model the observable behavior of a linguist performing *analysis by reduction* of Czech sentences on the blackboard or on a sheet of paper.

[1] Note that the layer of surface syntax does not correspond to any layer present in the theoretical specification of FGD, but rather to the auxiliary 'analytical' layer of the Prague Dependency Treebank, see Mikulová *et al.* (2005), which is technically useful for a maximal articulation of the process of analysis.

2. Analysis by Reduction for FGD

In this section we focus on the analysis by reduction for Functional Generative Description. After a brief characterization of analysis by reduction (subsection 2.1.), we address two basic linguistic phenomena, dependency (subsection 2.2.) and word order (2.3.), and illustrate the process of the analysis by reduction on examples from Czech.

2.1. Analysis by Reduction

The analysis by reduction makes it possible to formulate the relationship between dependency and word order (see also Lopatková *et al.* 2005). This approach is indispensable especially for modeling the syntactic structure of languages with a high degree of 'free' word order, where the dependency (predicate-argument) structure and word order are very loosely related. The restarting automaton M_{FGD} that models analysis by reduction for FGD is specified in detail in the Section 3.

The *analysis by reduction* is based on a stepwise simplification of a sentence – each step of analysis by reduction consists of deleting at least one word of the input sentence (see Lopatková *et al.* 2005 for more details).[2] The following principles must be satisfied:

- preservation of syntactic correctness of the sentence;
- preservation of the lemmas and sets of morphological categories;
- preservation of the meanings/senses of the words in the sentence (represented e.g. as an entry in a (valency) lexicon);
- preservation of the 'completeness' of the sentence (in this text only valency complementations (i.e., its arguments/inner participants and those of its adjuncts/free modifications that are obligatory) of frame evoking lexical items must be preserved).

The analysis by reduction works on a sentence (string of tokens) enriched with metalanguage categories from all the layers of FGD – in addition to word forms and punctuation marks, it embraces also morphological, surface and tectogrammatical information.

The input sentence is simplified until the so called *core predicative structure* of the sentence is reached. The core predicative structure consists of:

- the governing verb (predicate) of an independent verbal clause and its valency complementations, or
- the governing noun of an independent nominative clause and its valency complementations, e.g., *Názory čtenářů.* [Readers' opinions.], or

[2] Here we work only with the deleting operation whereas in Lopatková *et al.* (2005) the rewriting operation is also presupposed.

- the governing word of an independent vocative clause, e.g., *Jano!* [Jane!], or
- the governing node of an independent interjectional clause, e.g., *Pozor!* [Attention!].

2.2. Processing dependencies

Czech is a language with a high degree of so-called free word order. Naturally, (surface) sentences with permuted word order are not totally synonymous (as the word order primarily reflects the topic-focus articulation in Czech), but their grammaticality may not be affected and the dependency relations (as binary relations between governing and dependent lexical items) may be preserved regardless of the word order changes. This means that the identification of a governing lexical item and its particular complementations is not based primarily on their position in the sentence but rather on the possible order of their reductions.

There are two ways of processing dependencies during the analysis by reduction.

- Free modifications (i.e., adjuncts) that do not satisfy valency requirements of any lexical item in the sentence are deleted one after another, in an arbitrary order (sentence (1)).
- The so called reduction components (formed by words that must be reduced together to avoid non-grammaticality, i.e., incompleteness of tectogrammatical representation)[3] are processed 'en bloc' depending on their function in the sentence:
  - Either all members of the reduction component are reduced – this step is applied if the 'head' of the reduction component does not fulfill any valency requirements of any lexical item in the sentence (see sentence (2) below where the whole component represents optional free modification).
  - Or (if the 'head' of the reduction component satisfies the valency frame of some lexical item):
    (i) the item representing the 'head' is simplified – all the symbols apart from the functor[4] are deleted; the result of such a simplification can be understood as a zero lexical realization of the respective item, see sentence (3) below;

[3] Typically, a reduction component is composed of a frame evoking lexical item together with its valency complementations, see Lopatková *et al.* (2005). Let us stress here that a reduction component may constitute a discontinuous string.

[4] A functor is the label for syntactico-semantic relation holding between the respective item and its governing lexical item.

(ii) the complementation(s) of the `head' of the reduction component is/are deleted.

Convention: For the sake of clarity we have adopted the following conventions for displaying examples:

- Each column contains a symbol from one part of the (partitioned) vocabulary, that means information on one layer of FGD:[5]
  - the first column contains tokens,
  - the second column contains morphological lemmas (m-lemmas) and morphemic values (i.e., morphological categories),
  - the third column contains (surface) syntactic functions,
  - for autosemantic words,[6] the fourth column contains tectogrammatical lemmas (t-lemmas), functors, frame identifiers and other tectogrammatical categories (so called grammatemes).
- Each individual token and its metalanguage categories are located:
  - in one line if its surface word order position agrees with the deep word order (i.e., word order at the tectogrammatical layer), or the token has no tectogrammatical representation (i.e., it is not an autosemantic word);
  - in two lines if its surface word order position disagrees with the deep word order:

    (i) one line embraces the token, its m-lemma and morphemic values as well as its (surface) syntactic function, and

    (ii) the other line contains relevant tectogrammatical information (for autosemantic words).
- The top-down ordering of lines reflects the word order on the respective layer.

Such a two-dimensional convention allows for revealing both (i) a representation of a whole sentence on particular layers (individual columns for particular layers), including relevant word order (columns 1, 2, 3 reflect the surface word order whereas column 4 is organized according to deep word order), and (ii) information relevant for individual tokens (rows).

Let us illustrate the processing of dependencies on the examples.

Example:

(1) *Včera přišel domů pozdě.*
yesterday came home late
"Yesterday he came home late."

[5] The standard notation used in the Prague Dependency Treebank is used, see Hajič (2005).

[6] Function words have just functors or grammatemes as their tectogrammatical correlates that are assigned to their governing autosemantic words.

The analysis by reduction starts with the input string specified in Fig. 1. (see the convention above; the metalanguage categories are explained e.g. in Hajič 2005).[7]

Včera	*m-včera*.Dg- - -	Adv	*t-včera*.TWHEN
			[on].ACT
přišel	*m-přijít*.VpYS-	Pred	*t- přijít*.PRED.Framel.ind-ant
domů	*m-domů*.Db- - -	Adv	*t-domů*.DIR3
pozdě	*m-pozdě*.Dg- - -	Adv	*t-pozdě*.TWHEN
.	..Z: - - -	AuxK	

Fig. 1. The input string for sentence (1).

It is obvious that an item of TR (an autosemantic word, see for Note 6) can have zero surface lexical realization (e.g., actor, ACT need not be realized, as Czech is a pro-drop language – the corresponding item is restored in the TR; also different kinds of ellipsis are possible). On the other hand, several word forms can constitute a single item of TR (as e.g., a prepositional group in sentence (2)).

Let us point out the difference between the two types of free modifications in the sentence, namely DIR3 (direction `to_where') and TWHEN (temporal relation `when'): (i) whereas the valency complementation of direction DIR3 is considered to be obligatory for the verb *přijít* [to come] (the speaker as well as the listener must know this, see the dialogue test proposed in Panevová 1974) and thus fills the relevant slot of the valency frame of the verb (here marked by the label Frame1), (ii) the temporal relation TWHEN is an optional free modification (not belonging to the valency frame Frame1).

(2 steps) →

			[on].ACT
přišel	*m-přijít*.VpYS-	Pred	*t- přijít*.PRED.Framel.ind-ant
domů	*m-domů*.Db- - -	Adv	*t-domů*.DIR3
.	..Z: - - -	AuxK	

Fig. 2. The reduced string – a core predicative structure for sentence (2).

The first step of analysis by reduction consists in the deletion of one of the optional free modifications *včera* [yesterday] or *pozdě* [late].[8] These free

[7] We leave aside the problems of word order – this domain is briefly addressed in the following subsection.

[8] More precisely, the tokens as well as all the metalanguage categories relevant for the particular lexical item are reduced, similarly in the sequel.

modifications may be reduced in an arbitrary order, they are mutually independent (see Lopatková *et al.* 2005). These reduction steps result in the string in Fig. 2.

Now, the sentence contains only one reduction component constituted by the finite verb and its valency complementations, i.e., its actor (expressed by a zero form of the pronoun) and its obligatory free modification DIR3 'to_where', *[on] přišel domů* [(he) came home]. This is a core predicative structure, thus the reduction ends successfully.

Example:

(2) *Petr včera přišel do školy, kterou loni postavil*
Peter yesterday came to school which last_year built
minulý starosta.
previous major
"Yesterday Peter came to the school which was built last year by the
previous mayor."

This example shows the reduction of the whole reduction component that consists of a dependent clause. The input string looks as in Fig. 3.

Petr	*m-Petr*.NNMS1	Sb	*t-Petr*.ACT
včera	*m-včera*.Dg- - -	Adv	*t-včera*.TWHEN
přišel	*m-přijít*.VpYS-	Pred	*t- přijít*.PRED.Framel.ind-ant
do	*m-do*.RR- - 2	AuxP	
školy	*m-škola*.NNFS2	Adv	*t-škola*.DIR3.basic
,	,.Z: - - -	AuxK	
kterou	*m-který*.P4FS4	Obj	*t-který*.PAT
loni	*m-loni*.Db- - -	Adv	*t-loni*.TWHEN
postavil	*m-postavit*.VpYS-	Atr	*t-postavit*.RSTR.Frame2.ind-ant
minulý	*m-minulý*.AAMS1	Atr	
starosta	*m-starosta*.NNMS1	Sb	*t-starosta*.ACT
			t-minulý.RSTR
.	..Z: - - -	AuxK	

Fig. 3. The input string for sentence (2).

In the first three steps, the three optional free modifications *včera*, *loni* and *minulý* [yesterday, last_year, previous] are deleted in arbitrary order.

Next, the whole component *kterou postavil starosta* [which the mayor built] consisting of the verb and its valency complementations is to be processed. As this component represents an optional adnominal free modification RSTR, it can be simply deleted without the loss of completeness.

After this step, only one reduction component *Petr přišel do školy* [Peter came to school] remains, which constitute a core predicative structure – the analysis by reduction ends successfully.

Example:

(3) *Petr pomáhal Marii uklízet zahradu.*
Peter helped Mary to clean garden
"Peter helped Mary to clean the garden."

In this example there is a valency complementation realized as an infinitive form of the verb *uklízet* [to clean] and its two valency complementations, *[ona]* [she] (non-expressed) and *zahradu* [garden].[9]

In order to obtain the core predicative structure, the following simplification of the reduction component is used: (i) the complementations *[ona]* [she] and *zahradu* [garden] of the head verb *uklízet* [to clean] are deleted and (ii) the word form *uklízet* [to clean] and all the categories relevant to this word form apart from its functor (here PAT, patient) are deleted – such a simplified item represents a (saturated) lexical item with zero morphemic form (and thus, the valency requirements remain satisfied).

This step results in the core predicative structure.

2.3. Word Order

A large effort has been devoted to clearing up the role of word order in so called free-word order languages, see e.g. Hajičová *et al.* (1998), Holan *et al.* (2000), Havelka (2005), and Hajičová (2006) for some of the most recent contributions for Czech.

Let us recall two basic principles for the tectogrammatical representation of FGD (see esp. Sgall *et al.* 1986 and Hajičová *et al.* 1998):

- The word order in TR (deep word order) reflects the topic-focus articulation – it corresponds to the scale of communicative dynamism (thus it may differ from the surface word order).
- The theoretical research assumes the validity of the principle of projectivity for TRs.

These two principles have important consequences for the analysis by reduction that models the transition from surface form of a sentence to its TR – the surface word order must be modified in order to obtain the deep word order (example (4)). This holds particularly for sentences with non-projective surface

[9] We leave aside the relation of control, i.e., a specific type of grammatical coreference between a complementation of a governing node and (non-expressed) subject of the infinitive verb.

structure (example (5)). It implies that the sentence representation must in general reflect two word orders, the surface and the deep one. Let us repeat here the adopted convention of displaying examples, particularly that for word order – whereas columns 1, 2, 3 depict surface word order, column 4, reflecting tectogrammatical representation, reveals the deep word order.

Example: (see Mikulová *et al.* (2006), Sectiom 10.3.1.)
(4) *Černý kocour se napil ze své misky.*
black tomcat *refl* drunk from his bowl
"The black tomcat drank from its bowl."

Let us concentrate here on the topic focus articulation (see esp. Hajičová *et al.* 1998 and the writings quoted there).

According to Mikulová *et al.* (2006), the most general guideline of representing deep word order in TR is the placing of nodes representing contextually bound expressions to the left from their governing node and the placing of nodes representing contextually non-bound expressions to the right from their governing node. The contextual boundness is described in the attribute `tfa', the values `c' (contrastive topic), `t' (contextually bound) and `f' (contextually non-bound) belong to the metalanguage categories in the tectogrammatical vocabulary. The input string for analysis is in Fig. 4 (the last category in the fourth column, divided by `_', reflects tfa).

Černý	*m-černý*.NNMS1	Atr	
kocour	*m-kocour*.NNMS1	Sb	*t-kocour*.ACT_t
			t-černý.RSTR_f
			[Gen].PAT_t
se	*m-se*.P7-X4	AuxR	
napil	*m-napít*.VpYS-	Pred	*t-napít_se*.PRED.Frame5_f
ze	*m-z*.RV- - 2	AuxP	
své	*m-svůj*.P8FS2	Atr	[PersPron].APP_t
misky	*m-miska*.NNFS2	Adv	*t-miska*.DIR1.basic_f
.	..Z: - - -	AuxK	

Fig. 4. The input string for sentence (4).

The actor, ACT *kocour*_t [tomcat] is contextually bound and it appears to the left of its governing verb *napil_se*_f [drank] in the surface; the contextually non-bound DIR1 complementation *misky*_f [bowl] is to the right of its governing verb; and the contextually bound *svůj*_t [his] is to the left from its governing word *miska*_f [bowl] as well – the surface word order agrees in these cases with the deep word order.

On the other hand, the modification *černý*_f [black] is contextually non-bound and it stands before its (bound) governing word *kocour*_t [tomcat] – here the surface word order disagrees with the deep word order. This is the reason why the ordering in the last column (with the tectogrammatical representation) does not replicate the ordering of other columns – the contextually bound modification *černý*_f [black] appears at the second position in the TR of the sentence (just behind the governing item *kocour*_t [tomcat]).

Now, the reduction phase can start, i.e., a stepwise simplification of the sentence according to the principles of analysis by reduction, during which the dependencies are treated and the core predicative structure is obtained, as it is described in the previous subsection.

Example: (see Sgall *et al.* 1986, p. 241)
(5) *Karla plánujeme poslat na rok do Anglie.*
Charles (we) plan to_send for year to England
"Charles we are planning to send for a year to England."
≈ As for Charles, we are planning to send him for a year to England.

The proper noun *Karla*_c [Charles], which is the contrastive topic of a sentence (tfa = `c'), is moved away from its governing verb *poslat*_f [to send], which causes a non-projectivity in the surface structure. The theoretical assumption of projectivity of TRs requires a different deep order – the corresponding item *t-Charles*.PAT_c in TR is situated just before its governing item *t-poslat*.PRED.Frame1_f [to send].

The analysis by reduction has the input string as in Fig. 5.

Karla	*m-Karel*.NNMS4	Obj	
			[my].ACT_t
plánujeme	*m-plánovat*.VB-P-	Pred	*t-plánovat*.PRED.Frame6.ind-sim_f
			t-Karel.PAT_c
			[my].ACT_t
poslat	*m-poslat*.Vf- - -	Obj	*t-poslat*.PAT.Frame7_f
na	*m-na*.RR- - 4	AuxP	
rok	*m-rok*.NNIS4	Adv	*t-rok*.THL_f
do	*m-do*.RR- - 2	AuxP	
Anglie	*m-Anglie*.NNFS2	Adv	*t-Anglie*.DIR3.basic_f
.	..Z: - - -	AuxK	

Fig. 5. The input string for sentence (5).

Now, the reduction phase treating the dependencies can start.

3. The *4-LRL*-automata

In this section, the formal model for analysis by reduction for FGD is proposed. We use here the standard way of presentation from the theory of automata (our remarks should hopefully help readers not quite familiar with that kind of presentation). This section is partitioned into two subsections. The first one introduces *sRL-automata* – the basic models of restarting automata we will be dealing with. The important notion of metarules is introduced here; they serve for a more transparent, more declarative description of restarting automata.

The second subsection introduces *4-LRL-automata* as a special case of *sRL*-automata. A four-level *analysis by reduction system*, which is an algebraic representation of analysis by reduction, and the formal languages which represent the individual layers of FGD are introduced here, namely the languages of the first and the last level that correspond to the surface language *LC* and to the tectogrammatical language *LM* from Section 1. Further, the *characteristic relation SH(M)* is introduced.

Finally, the *SH-synthesis*, which models FGD as a generative device and specifies the generative ability of FGD, and *SH-analysis*, which fulfills the task of syntactico-semantic analysis of FGD, are introduced here step by step.

3.1. The *t-sRL*-Automaton

Here we describe in short the type of restarting automaton we will be dealing with. The subsection is an adapted version of the first part of Messerschmidt *et al.* (2006). More (formal) details of the development of restarting automata can be found in Otto (2006).

An *sRL-automaton* (*simple RL-automaton*) *M* is (in general) a nondeterministic machine with a finite-state control *Q*, a finite characteristic vocabulary Σ, and a head with the ability to scan exactly one symbol (word) that works on a flexible tape delimited by the left sentinel ¢ and the right sentinel $.

Let us proceed a bit more formally. A simple *RL-automaton* is a tuple $M = (Q, \Sigma, \delta, q_0, ¢, \$)$, where:

- Q is a finite set of states,
- Σ is a finite vocabulary (the characteristic vocabulary),
- *¢, $* are sentinels, *{¢, $}* do not belong to Σ,
- q_0 from Q is the initial state,
- δ is the transition relation ≈ a finite set of instructions of the shape (q,a) $\rightarrow_M$ *(p,Op)*, where *q, p* are states from *Q*, *a* is a symbol from Σ, and *Op* is an operation, where the particular operations correspond to the particular types of steps (move-right, move-left, delete, accept, reject, and restart step).

For an input sentence $w \in \Sigma^*$, 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 *move-right steps* and *move-left steps*, which change the state of M and shift the window one position to the right or to the left, respectively, and *delete steps*, which delete the content of the window, thus shorten the tape, change the state, and shift the window to the right neighbor of the symbol deleted. Of course, neither the left sentinel $¢$ nor the right sentinel $\$$ may 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 before the first restart step and also between any two restart steps, M executes at least one delete operation.

A *configuration* of M is a string $\alpha q \beta$ where $q \in Q$, and either $\alpha = \lambda$ and $\beta \in \{¢\} \cdot \Sigma^* \cdot \{\$\}$ or $\alpha \in \{¢\} \cdot \Sigma^*$ and $\beta \in \Sigma^* \cdot \{\$\}$; here q represents the current state, $\alpha\beta$ is the current content of the tape, and it is understood that the window contains the first symbol of β. A configuration of the form $q_0¢w\$$ is called a *restarting configuration*.

We observe that each computation of an *sRL*-automaton M consists of certain phases. Each part of a computation of M from a restarting configuration to the next restarting configuration is called a *cycle*. The part after the last restart operation is called the *tail*. We use the notation $u \vdash_M^c v$ to denote a cycle of M that begins with the restarting configuration $q_0¢u\$$ and ends with the restarting configuration $q_0¢v\$$; the relation $\vdash_M^{c^*}$ is the reflexive and transitive closure of $\vdash_M^c$.

An input $w \in \Sigma^*$ is *accepted by M*, if there is an accepting computation which starts with the (initial) configuration $q_0¢w\$$. By $L_\Sigma(M)$ we denote the *characteristic language* consisting of all strings accepted by M; we say that M *recognizes (accepts) the language $L_\Sigma(M)$*. By $S_\Sigma(M)$ we denote the *simple language* accepted by M, which consists of all strings that M accepts by computations without a restart step. By *sRL* we denote the class of all sRL-automata.

A *t-sRL-automaton* ($t \geq 1$) is an *sRL*-automaton M which uses at most t delete operations in a cycle and any string of $S_\Sigma(M)$ has no more than t symbols (wordforms).

Remark: The *t-sRL*-automata are two-way automata which allow, in any cycle, to check the whole sentence before reduction (deleting). This reminds us of the behavior of a linguist who can read the whole sentence before choosing the reduction. The automaton should be non-deterministic in general in order to be able to change the order of deleting cycles. That serves for witnessing the independence of some parts of the sentence, see the section about the analysis by reduction. Another message from this section is that there is a t which creates a boundary for the number of deletions in a cycle and for the size of the accepted irreducible strings.

Based on Messerschmidt *et al.* (2006), we can describe a t-sRL-automaton by *metainstructions* of the form

$(¢ \cdot E_0, a_1, E_1, a_2, E_2, ..., E_{s-1}, a_s, E_s \cdot \$), 1 \leq s \leq t$, where

- $E_0, E_1, ..., E_s$ are regular languages (often represented by regular expressions), called the *regular constraints* of this instruction, and
- $a_1, a_2, …, a_s \in \Sigma$ correspond to letters that are deleted by M during one cycle.

In order to execute this metainstruction, M starts from a configuration $q_0¢w\$$; it will get stuck (and so reject), if w does not admit a factorization of the form $w = v_0a_1v_1a_2...v_{s-1}a_sv_s$ such that $v_i \in E_i$ for all $i = 0, ..., s$. On the other hand, if w admits factorizations of this form, then one of them is chosen nondeterministically, and the restarting configuration $q_0¢w\$$ is transformed into $q_0¢v_0v_1…v_{s-1}v_s\$$. To describe also the tails of the accepting computations, we use accepting metainstructions of the form $(¢ \cdot E \cdot \$, Accept)$, where E is a regular language (finite in this case). Moreover, we can require that there is only a single accepting metainstruction for M.

Example: Let us illustrate the power of restarting automata on the formal language L_{Rt}. Let $t \leq 1$, and let $L_{Rt} = \{c_0wc_1wc_2...c_{t-1}w \ / w \in \{a,b\}^*\}$. For this language, a *t-sRL*-automaton M_t with a vocabulary $\Sigma_t = \{c_0,c_1,...,c_{t-1}\} \cup \Sigma_0$, where $\Sigma_0 = \{a,b\}$, can be obtained through the following sequence of metainstructions:

(1) $(¢c_0, a, \Sigma_0^* \cdot c_1, a , \Sigma_0^* \cdot c_2, ..., \Sigma_0^* \cdot c_{t-1}, a, \Sigma_0^* \cdot \$)$,

(2) $(¢c_0, b, \Sigma_0^* \cdot c_1, b , \Sigma_0^* \cdot c_2, ..., \Sigma_0^* \cdot c_{t-1}, b, \Sigma_0^* \cdot \$)$,

(3) $(¢c_0 ... c_{t-1}\$, Accept)$.

It follows easily that $L(M_t) = L_{Rt}$ holds.

We emphasize the following property of restarting automata. It plays an important role in our applications of restarting automata.

Definition (Correctness Preserving Property)

A *t-sRL*-automaton M is *(strongly) correctness preserving* if $u \in L_\Sigma (M)$ and $u \vdash_M^{c^*} v$ imply that $v \in L_\Sigma (M)$.

It is rather obvious that all deterministic *t-sRL*-automata are correctness preserving. On the other hand, one can easily construct examples of nondeterministic *t-sRL*-automata that are not correctness preserving.

3.2. The *4-LRL*-automata and related notions

Let us finally introduce the model of automaton proposed for modeling of analysis by reduction for FGD. A *4-LRL-automaton* (*4-level sRL-automaton*) M_{FGD} is a correctness preserving *t-sRL*-automaton. Its characteristic vocabulary Σ is partitioned into four subvocabularies $\Sigma_0, ..., \Sigma_3$. M_{FGD} deletes at least one symbol from Σ_0 in each cycle.

Remark: The correctness preserving property of M_{FGD} ensures a good simulation of the linguist performing the analysis by reduction. Similarly as the linguist, the automaton M_{FGD} should not make a mistake during analysis by reduction, otherwise there is something wrong, e.g., the characteristic language is badly proposed. This situation can be fixed by adding some new categories (symbols). The correctness preserving property can be automatically tested. This may be useful for checking and improving a language description in the context of FGD. The request of the deletion of at least one surface wordform in any cycle represents the request of the (generalized) lexicalization of FGD.

Let us inherit the notion $L_\Sigma(M_{FGD})$, characteristic language of M_{FGD}, and $S_\Sigma(M_{FGD})$, the simple language, from the previous subsection. All the notions introduced below are derived from these notions.

As the first step, we introduce an *(analysis by) reduction system* involved by M_{FGD}, and by the set of level alphabets $\Sigma_0, ..., \Sigma_3$. It is defined as follows:

$RS(M_{FGD}) = (\Sigma^*, \vdash_{MFGD}^{c}, S_\Sigma(M_{FGD}), \Sigma_0, ..., \Sigma_3)$.

The reduction system (by M_{FGD}) formalizes the notion of the analysis by reduction of FGD in an algebraic, non-procedural way. Observe that for each $w \in \Sigma^*$ we have $w \in L_\Sigma(M_{FGD})$ if and only if $w \vdash_{MFGD}^{c*} v$ holds for some string $v \in S_\Sigma(M_{FGD})$.

A *language of level j recognized by* M_{FGD}, where $0 \leq j \leq 3$, is the set of all sentences (strings) that are obtained from $L_\Sigma(M_{FGD})$ by removing all symbols which do not belong to Σ_j. We denote it $L_j(M_{FGD})$. Particularly, $L_0(M_{FGD})$ represents the surface language *LC* defined by M_{FGD}; similarly, $L_3(M_{FGD})$ represents the language of tectogrammatical representations *LM* defined by M_{FGD} (see Section 1).

Now we can define the *characteristic relation* $SH(M_{FGD})$ given by M_{FGD}:

$SH(M_{FGD}) = \{(u,y) \mid u \in L_0(M_{FGD}),\ y \in L_3(M_{FGD})$ and there is a $w \in L_\Sigma(M_{FGD})$ such that u is obtained from w by deleting the symbols not belonging to Σ_0, and y is obtained from w by deleting the symbols not belonging to Σ_3 $\}$.

Remark: The characteristic relation represents the basic relations in language description, relations of synonymy and ambiguity in language *L*. In other words, it embraces the translation of the surface language *LC* into the tectogrammatical language and vice versa. From this notion, the remaining notions, analysis and synthesis, can be derived.

We introduce the *SH-synthesis by* M_{FGD} for any $y \in LM$ as a set of pairs *(u,y)* belonging to $SH(M_{FGD})$:

$synthesis\text{-}SH(M_{FGD},y) = \{(u,y) \mid (u,y) \in SH(M_{FGD})\}$

The *SH*-synthesis associates a tectogrammatical representation (i.e., string *y* from *LM*) with all its possible surface sentences *u* belonging to *LC*. This notion allows for checking the synonymy and its degree provided by M_{FGD}. The linguistic issue is to decrease the degree of the synonymy by M_{FGD} by the gradual refinement of M_{FGD}.

Finally we introduce the dual notion to the *SH*-synthesis, the *SH-analysis by* M_{FGD} *of* $u \in LC$:

$analysis\text{-}SH(M_{FGD},u) = \{(u,y) \mid (u,y) \in SH(M_{FGD})\}$

The *SH*-analysis returns, to a given surface sentence u, all its possible tectogrammatical representations, i.e., it allows for checking the ambiguity of an individual surface sentence. This notion provides the formal definition for the task of full syntactico-semantic analysis by M_{FGD}.

4. Concluding remarks

The paper presents the basic formal notions that allow for formalizing the notion of analysis by reduction for Functional Generative Description, FGD. We have outlined and exemplified the method of analysis by reduction and its application in processing dependencies and word order in a language with a high degree of free word order. Based on this experience, we have introduced the 4-level reduction system for FGD based on the notion of simple restarting automata. This new formal frame allows us to define formally the characteristic relation for FGD, which renders synonymy and ambiguity in the studied language.

Such a formalization makes it possible to propose a software environment for the further development. It provides a possibility to describe exactly the basic phenomena observed during linguistic research. Further, it allows for studying suitable algorithms for tasks in computational linguistics, namely automatic syntactico-semantic analysis and synthesis.

The presented notions are also useful to show exactly the differences and similarities between the methodological basis of our (computational) linguistic school and the methodological bases of other schools. The basic message given here is to show the possibility of generalizing the principle of lexicalization trough the layers in order to obtain a checking procedure for FGD via analysis by reduction.

References

Hajič, Jan. 2005. Complex Corpus Annotation: The Prague Dependency Treebank. In *Insight into Slovak and Czech Corpus Linguistics*, ed. M. Šimková, 54–7 3. Veda Bratislava.

Hajičová, Eva, Barbara H, Partee, and Petr Sgall. 1998. *Topic-Focus Articulation, Tripartite Structures, and Semantic Content.* Kluwer, Dordrecht.

Hajičová, Eva. 2006. K některým otázkám závislostní gramatiky. *Slovo a slovesnost* 67:3–26.

Havelka, Jiří. 2005. Projectivity in Totally Ordered Rooted Trees. *The Prague Bulletin of Mathematical Linguistics* 84:13–30.

Holan, Tomáš, Vladislav Kuboň, Karel Oliva, and Martin Plátek. 2000. On Complexity of Word Order. In *Les grammaires de dépendance – Traitement automatique des langues*, ed. S. Kahane,Vol. 41, No. 1, 273–300.

Lopatková, Markéta, Martin Plátek, and Vladislav Kuboň. 2005. Modeling Syntax of Free Word-Order Languages: Dependency Analysis by Reduction. In *Lecture Notes in Computer Science*, Vol. 3658, 140–147.

Messerschmidt Hartmut, František Mráz, Friedrich Otto, and Martin Plátek. 2006. Correctness Preservation and Complexity of Simple RL-Automata. In *Lecture Notes in Computer Science*, Vol. 4094, 162–172.

Mikulová, Marie *et al.* 2006. Annotation on the Tectogrammatical Layer in the Prague Dependency Treebank. Technical report, Prague, MFF UK.

Otto, Friedrich. 2006. Restarting Automata. In: Recent Advances in Formal Languages and Applications. ed. Z Ésik, C. Martin-Vide, V. Mitrana), In *Studies in Computational Intelligence*, Vol. 25, 269–303. Springer, Berlin.

Panevová, Jarmila. 1974. On Verbal Frames in Functional Generative Description. *The Prague Bulletin of Mathematical Linguistics* 22, 3–40.

Petkevič, Vladimír. 1995. A New Formal Specification of Underlying Structure. *Theoretical Linguistics* Vol.21, No.1.

Plátek, Martin: 1982. Composition of Translation with D-trees. In *COLING' 82*, 313–318.

Plátek, Martin, and Petr Sgall. 1978. A Scale of Context-Sensitive Languages: Applications to Natural Language. *Information and Control*. Vol. 38., No 1, 1-20.

Sgall, Petr, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects.* Ed. J. Mey, Dordrecht: Reidel and Prague: Academia.

Sgall Petr, Ladislav Nebeský, Alla Goralčíková, and Eva Hajičová. 1969. *A Functional Approach to Syntax in Generative Description of Language.* New York.

Markéta Lopatková, Martin Plátek, Petr Sgall
Charles University in Prague
Malostranské nám. 25
Prague 1, 118 00, Czech Republic
lopatkova@ufal.mff.cuni.cz