PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 87 JUNE 2007 7–26

Towards a Formal Model for Functional Generative Description

Analysis by Reduction and Restarting Automata

Markéta Lopatková, Martin Plátek, Petr Sgall

Abstract

Functional Generative Description (FGD) is a dependency based descriptive system, which has been in development since the 1960s, see esp. Sgall et al. (1969). FGD was originally implemented as a generative procedure, but lately we have been interested in a declarative representation. 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 a new formal frame, namely a *4-level reduction system* for FGD, which is based on the notion of *simple restarting automata*, see Messerschmidt et al. (2006). This new approach mirrors straightforwardly the so-called *(multi-level) analysis by reduction*, an implicit method used for linguistic research – analysis by reduction allows for obtaining (surface and/or deep) (in) dependencies by the reductions of Czech sentences as well as for describing properly the complex word order of a free word order language, see Lopatková, Plátek, and Kuboň (2005).

1. Introduction

Functional Generative Description (FGD) is a dependency based system for Czech, which has been in development since the 1960s, see esp. Sgall et al. (1969); Sgall, Hajičová, and Panevová (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

Please cite this article as: Markéta Lopatková, Martin Plátek, Petr Sgall, Towards a Formal Model for Functional Generative Description: Analysis by Reduction and Restarting Automata. The Prague Bulletin of Mathematical Linguistics No. 87, 2007, 7–26.

intensional semantics, see Hajičová, Partee, and Sgall (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 and Sgall (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:

1. The set of correct sentences of the language L, denoted by LC.
2. The formal language LM representing all possible tectogrammatical representations (TRs) of sentences in L.
3. The relation SH between LC and LM describing the ambiguity and the synonymy of L.
4. 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*).

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 *analysis by reduction*. Analysis by reduction 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, Plátek, and Kuboň (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. 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 – section 2.1 provides a brief characterization of analysis by reduction.

Example: The example presented in Fig. 1 outlines the form of the input for analysis by reduction used in this paper, demonstrated on the sentence (1):

(1) *Přišel domů pozdě.*
[came home late]
E. He came home late.

There are four (sub)vocabularies Σ_0, Σ_1, Σ_2, Σ_3, each subvocabulary Σ_i represents the

			[on].ACT
Přišel	*m-přijít*.VpYS-	Pred	*t-přijít*.PRED.Frame1.ind-ant
domů	*m-domů*.Db- - -	Adv	*t-domů*.DIR3
pozdě	*m-pozdě*.Dg- - -	Adv	*t-pozdě*.TWHEN
.	..Z: - - -	AuxK	

Figure 1. A sample input structure for analysis by reduction for sentence (1).

corresponding layer of language description in FGD, namely:[1]

- 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);[2]
- 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).

In Section 2 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 modelling 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, and a head (window of size 1) that works on a flexible tape. 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, and
- *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.

[1]The first column in the figure contains symbols from a vocabulary Σ_0, the second one contains symbols from a vocabulary Σ_1 and so on, the convention for displaying examples is specified in Section 2.2.

[2]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 Hajič (2005), which is technically useful for a maximal articulation of the process of analysis.

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 $\Sigma_0, \cdots, \Sigma_3$.

The basic notion related to M_{FGD} is the notion of the language accepted by M_{FGD}, so called *characteristic language* $L_C(M_{FGD})$. 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 $\Sigma_1, \Sigma_2, \Sigma_3$. The tectogrammatical language LM as well as the relation SH can be extracted from $L_C(M_{FGD})$.

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.

2. Analysis by reduction for FGD

In this section we focus on the analysis by reduction for Functional Generative Description. 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, Plátek, and Kuboň (2005). This approach is indispensable especially for modelling 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, Plátek, and Kuboň (2005) for more details.[3] 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.

[3]Here we work only with the deleting operation whereas in Lopatková, Plátek, and Kuboň (2005) the rewriting operation also is presupposed.

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
- 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 (2)).
- The so called reduction components (formed by words that must be reduced together to avoid non-grammaticality, i.e. incompleteness of tectogrammatical representation)[4] 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 sentences (3) and (5) below where the whole components represent optional adnominal free modifications).
  - Or (if the 'head' of the reduction component satisfies the valency frame of some lexical item):
    1. the item representing the 'head' is simplified – all the symbols apart from the functor[5] are deleted; the result of such a simplification can be understood as a zero lexical realization of the respective item, see sentence (4); and
    2. 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:

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

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

- Each column contains a symbol from one part of the (partitioned) vocabulary, that means information on one layer of FGD:[6]
  - 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,[7] 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 'separate' 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:
    1. one line embraces the token, its m-lemma and morphemic values as well as its (surface) syntactic function, and
    2. 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 reflects 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 sentences (2), (3), (4) and (5).

Example:

(2) *Včera přišel domů pozdě.*
[yesterday came home late]
E. Yesterday he came home late.

The analysis by reduction starts with the input structure specified in Fig. 2 (see the convention above; the metalanguage categories are explained e.g. in Hajič, 2005).

It is obvious that an item of TR (an autosemantic word, see for Note 7) 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 (3)).

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

[6] Here the standard annotation used in the Prague Dependency Treebank is used, see Hajič (2005).

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

Včera	*m-včera*.Dg- - -	Adv	*t-včera*.TWHEN
			[on].ACT
přišel	*m-přijít*.VpYS-	Pred	*t-přijít*.PRED.Frame1.ind-ant
domů	*m-domů*.Db- - -	Adv	*t-domů*.DIR3
pozdě	*m-pozdě*.Dg- - -	Adv	*t-pozdě*.TWHEN
.	..Z: - - -	AuxK	

Figure 2. The input structure for sentence (2).

(2 steps) ⟶

			[on].ACT
přišel	*m-přijít*.VpYS-	Pred	*t-přijít*.PRED.Frame1.ind-ant
domů	*m-domů*.Db- - -	Adv	*t-domů*.DIR3
.	..Z: - - -	AuxK	

Figure 3. The reduced structure – a core predicative structure for sentence (2).

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 *přijít* [to come] (here marked by the label Frame1), (ii) the temporal relation TWHEN is an optional free modification (not belonging to the valency frame Frame1).

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 modifications may be reduced in an arbitrary order, they are mutually independent, see Lopatková, Plátek, and Kuboň (2005). These two reduction steps result in the structure in Fig. 3.

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.[9]

Example: This example shows the reduction of the whole reduction component that consists of a dependent clause.

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

The input structure looks as in Fig. 4.

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

[9] Here we leave aside the problems of word order – this domain is briefly addressed in the following subsection.

Petr	*m-Petr*.NNMS1	Sb	*t-Petr*.ACT
včera	*m-včera*.Dg- - -	Adv	*t-včera*.TWHEN
přišel	*m-přijít*.VpYS-	Pred	*t-přijít*.PRED.Frame1.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	

Figure 4. The input structure for sentence (3).

(3 steps) →

Petr	*m-Petr*.NNMS1	Sb	*t-Petr*.ACT
přišel	*m-přijít*.VpYS-	Pred	*t-přijít*.PRED.Frame1.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
postavil	*m-postavit*.VpYS-	Atr	*t-postavit*.RSTR.Frame2.ind-ant
starosta	*m-starosta*.NNMS1	Sb	*t-starosta*.ACT
.	..Z: - - -	AuxK	

Figure 5. The simplified structure for sentence (3).

In the first three steps, the three optional free modifications *včera, loni* and *minulý* [yesterday, last_year, previous] are deleted in arbitrary order, see Fig. 5.

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, see Fig. 6 which constitute a core predicative structure – the analysis by reduction ends successfully.

Example: Let us show an analysis of a sentence with a valency complementation realized as an infinitive form of the verb.

→

Petr	*m-Petr*.NNMS1	Sb	*t-Petr*.ACT
přišel	*m-přijít*.VpYS-	Pred	*t-přijít*.PRED.Frame1.ind-ant
do	*m-do*.RR- - 2	AuxP	
školy	*m-škola*.NNFS2	Adv	*t-škola*.DIR3.basic
.	..Z: - - -	AuxK	

Figure 6. The core predicative structure for sentence (3).

Petr	*m-Petr*.NNMS1	Sb	*t-Petr*.ACT
pomáhal	*m-pomáhat*.VpYS-	Pred	*t-pomáhat*.PRED.Frame1.ind-ant
Marii	*m-Marie*.NNFS3	Obj	*t-Marie*.ADDR
			[ona].ACT
uklízet	*m-uklízet*.Vf- - -	Adv	*t-uklízet*.PAT.Frame3
zahradu	*m-zahrada*.NNFS4	Obj	*t-zahrada*.PAT
.	..Z: - - -	AuxK	

Figure 7. The input structure for sentence (4).

(4) *Petr pomáhal Marii uklízet zahradu.*
[Peter helped Mary clean garden]
E. Peter helped Mary to clean the garden.

In this sentence 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],[10] see Fig. 7.

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 in Fig. 8.

Example: The following construction (called genitive of property, see Šmilauer, 1966, p. 175) is another example of reduction component.

[10] We leave aside the relation of control, i.e. a specific type of grammatical coreference between a complementation of a governing node, called controller – here *Marie* as ADDR (addressee) of the verb *pomáhat* [to help] – and (non-expressed) subject of the infinitive verb, called controllee – here *uklízet* [to clean].

→

Petr	*m-Petr*.NNMS1	Sb	*t-Petr*.ACT
pomáhal	*m-pomáhat*.VpYS-	Pred	*t-pomáhat*.PRED.Frame1.ind-ant
Marii	*m-Marie*.NNFS3	Obj	*t-Marie*.ADDR
			[ ].PAT
.	..Z: - - -	AuxK	

Figure 8. The core predicative structure for sentence (4).

			[on].ACT
Uviděl	*uvidět*.VpYS-	Pred	*t-uvidět*.PRED.Frame4.ind-ant
dívku	*m-dívka*.NNFS4	Obj	*t-dívka*.PAT
vysoké	*m-vysoký*.AAFS2	Atr	
postavy	*m-postava*.NNFS2	Atr	*t-postava*.APP
			t-vysoký.RSTR
.	..Z: - - -	AuxK	

Figure 9. The input structure for sentence (5).

(5) *Uviděl dívku vysoké postavy.*
[saw girl (of) tall figure]

E. He saw a girl with a tall figure.

The adnominal attribute (realized (usually) as a noun in genitive case), here *postavy* [figure], obligatorily requires some modification, here *vysoké* [tall], see Fig. 9.

This means that the whole component *vysoké postavy* [(with a) tall figure] must be processed within one cycle. As the head of the component *postavy* [figure] is not required by the valency of the verb, both parts of the reduction component are simply deleted in one cycle. Thus, the core predicative structure is obtained, see Fig. 10.

→

			[on].ACT
Uviděl	*uvidět*.VpYS-	Pred	*t-uvidět*.PRED.Frame4.ind-ant
dívku	*m-dívka*.NNFS4	Obj	*t-dívka*.PAT
.	..Z: - - -	AuxK	

Figure 10. The core predicative structure for sentence (5).

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, Partee, and Sgall (1998); Holan et al. (2000); Havelka (2005); 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, Hajičová, and Panevová (1986); Hajičová, Partee, and Sgall (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.[11]

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 (sentence (6)). This holds particulary for sentences with non-projective surface structure (sentence (7)). 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: Let us concentrate here on the topic focus articulation, see esp. Hajičová, Partee, and Sgall (1998) and the writings quoted there.

(6) *Černý kocour se napil ze své misky.* (see Mikulová et al., 2006, Section 10.3.1.)
[black tomcat *refl* drunk from its bowl]
E. The black tomcat drank from its bowl.

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 representations. The input structure for analysis is in Fig. 11, the last category in the fourth column, divided by '_', reflects tfa.

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

[11] A great number of definitions of projectivity appears in literature since the 1960s, more or less formal. In Mikulová et al. (2006) the projectivity is defined as follows: 'if two nodes M and N are connected by an edge and M is to the left from N, then all nodes to the right from M and to the left from N are connected with the root via a path that passes through at least one of the nodes M and N. In short: between a mother and its direct daughter there can be only direct or indirect daughters of the mother.'

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	

Figure 11. The input structure for sentence (6).

*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.[12]

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 is described in the previous subsection.

Example: Sentence (7) has non-projective surface realization.

(7) *Karla plánujeme poslat na rok do Anglie.*
[Charles plan to_send for year to England]
(see Sgall, Hajičová, and Panevová, 1986, p. 241)
E. 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 the 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 structure given in Fig. 12.

Now, the reduction phase treating the dependencies can start.

[12] We suppose that also restored ellipses (here [Gen].t_PAT, generalized adverbal patient, PAT) are placed in the respective position in the input string.

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	

Figure 12. The input structure for sentence (7).

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 $M = (Q, \Sigma, \delta, q_0, ¢, \$)$,

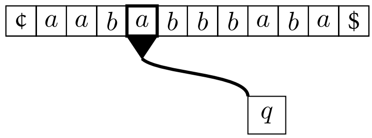

Figure 13. Restarting automaton

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 $\approx$ a finite set of instructions of the shape : $(q, a) \to_M (p, Op)$, 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).

- move right
- move left
- delete
- restart

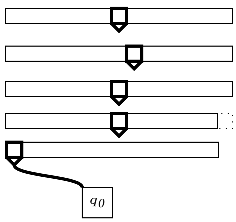

Figure 14. Operations

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

- restarting configuration
- accepting configuration

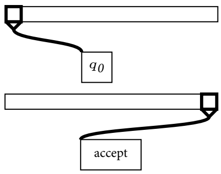

Figure 15. Basic configurations.

$\alpha \in \{¢\} \cdot \Sigma^*$ 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 $\vdash^{c^*}_M$ is the reflexive and transitive closure of $\vdash^c_M$.

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 $L_C(M)$ we denote the *characteristic language* consisting of all strings accepted by M; we say that M *recognizes (accepts) the language* $L_C(M)$. By $S_C(M)$ we denote the *simple language* accepted by M, which consists of all strings that M accepts by computations without a restart step. Obviously, $S_C(M)$ is a regular sublanguage of $L_C(M)$. By sRL we denote the class of all sRL-automata.

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

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

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

- $E_0, E_1, \ldots, E_s$ are regular languages (often represented by regular expressions), called the *regular constraints* of this instruction, and
- $a_1, a_2, \ldots, a_s \in \Sigma$ 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 $w = v_0 a_1 v_1 a_2 \cdots v_{s-1} a_s v_s$ such that $v_i \in E_i$ for all $i = 0, \dots, s$. On the other hand, if w admits factorizations of this form, then one of them is chosen nondeterministically, and the restarting configuration $q_0¢w\$$ is transformed into $q_0¢v_0 v_1 \cdots v_{s-1} v_s\$$. To describe also the tails of the accepting computations, we use accepting metainstructions of the form $(¢ \cdot E \cdot \$, \mathsf{Accept})$, 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 $t \geq 1$, and let $L_{R_t} = \{ c_0 w c_1 w c_2 \cdots c_{t-1} w \mid w \in \{a,b\}^* \}$. For this language, a t-sRL-automaton M_t with a vocabulary $\Sigma_t = \{c_0, c_1, \dots, c_{t-1}\} \cup \Sigma_0$, where $\Sigma_0 = \{a, b\}$, can be obtained through the following sequence of metainstructions:

$$\begin{array}{ll} (1) & (¢c_0, a, {\Sigma_0}^* \cdot c_1, a, {\Sigma_0}^* \cdot c_2, \dots, {\Sigma_0}^* \cdot c_{t-1}, a, {\Sigma_0}^* \cdot \$), \\ (2) & (¢c_0, b, {\Sigma_0}^* \cdot c_1, b, {\Sigma_0}^* \cdot c_2, \dots, {\Sigma_0}^* \cdot c_{t-1}, b, {\Sigma_0}^* \cdot \$), \\ (3) & (¢c_0 c_1 \cdots c_{t-1}\$, \mathsf{Accept}). \end{array}$$

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

We emphasize the following properties of restarting automata.

Definition: (Error Preserving Property) A t-sRL-automaton M is *error preserving* if $u \notin L_C(M)$ and $u \vdash^{c^*}_M v$ imply that $v \notin L_C(M)$.

The following property plays an important role in our applications of restarting automata.

Definition: (Correctness Preserving Property) A t-sRL-automaton M is *correctness preserving* if $u \in L_C(M)$ and $u \vdash^{c^*}_M v$ imply that $v \in L_C(M)$.

It is rather obvious that each t-sRL-automaton is error preserving, and 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 modelling of analysis by reduction for FGD. A 4-LRL-*automaton* (*4-level* sRL-*automaton*) M_{FGD} is a (correctness preserving) t-sRL-automaton, where its characteristic vocabulary Σ is composed from four subvocabularies $\Sigma_0, \dots, \Sigma_3$. M_{FGD} deletes at least one symbol from Σ_0 in each cycle.

Remark: The correctness and error preserving properties of M_{FGD} should ensure 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 improved 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 notions $L_C(M_{FGD})$, *characteristic language* of M_{FGD} and $S_C(M_{FGD})$, *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 $\Sigma_0, \dots, \Sigma_3$. It is defined as follows:

$$\mathsf{RS}(M_{FGD}) := (\Sigma^*, \vdash^c_{M_{FGD}}, S_C(M_{FGD}), \Sigma_0, \cdots, \Sigma_3).$$

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 $w \in L_C(M_{FGD})$ if and only if $w \vdash^{c^*}_{M_{FGD}} v$ holds for some string $v \in S_C(M_{FGD})$.

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 $L_C(M_{FGD})$ 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}) $= \{(u, y) \mid$ there is a $w \in L_C(M_{FGD})$ such that $u \in L_0(M_{FGD})$ and u is obtained from w by deleting the symbols not belonging to Σ_0, and $y \in L_3(M_{FGD})$ and y is obtained from w by deleting the symbols not belonging to $\Sigma_3\}$.

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 from* LM as a set of pairs (u, y) belonging to SH(M_{FGD}).

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

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_{FDG}. The linguistic issue is to decrease the degree of the synonymy by M_{FDG} by the gradual refinement of M_{FDG}.

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

$$analysis\text{-}\mathsf{SH}(M_{FGD}, u) = \{(u, y) | (u, y) \in \mathsf{SH}(M)_{FGD}\}$$

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

[*Včera*]$_1$	[*m-včera*.Dg- - -]$_2$	[Adv]$_3$	[*t-včera*.TWHEN]$_4$
			[[on].ACT]$_5$
[*přišel*]$_6$	[*m-přijít*.VpYS-]$_7$	[Pred]$_8$	[*t-přijít*.PRED.Frame1.ind-ant]$_9$
[*domů*]$_{10}$	[*m-domů*.Db- - -]$_{11}$	[Adv]$_{12}$	[*t-domů*.DIR3]$_{13}$
[*pozdě*]$_{14}$	[*m-pozdě*.Dg- - -]$_{15}$	[Adv]$_{16}$	[*t-pozdě*.TWHEN]$_{17}$
[.]$_{18}$	[..Z: - - -]$_{19}$	[AuxK]$_{20}$	

Figure 16. The input string for sentence (2).

M_{FDG}. The linguistic task is to refine M_{FDG} gradually, especially with respect to the description of ambiguity of the sentence.

Remark: Fig. 16 illustrates the transformation of the input structures used in Section 2 into the input strings for a M_{FDG} automaton. The individual numbered items in square brackets in Fig. 16 represents the individual symbols on the input tape of M_{FDG}. E.g., [*Včera*]$_1$ is the first symbol on the tape (after the left sentinel) belonging to Σ_0; [*m-včera*.Dg- - -]$_2$ is the second symbol (it is from Σ_1) and so on. [AuxK]$_{20}$ is the last symbol (item) on the tape before the right sentinel.

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.

Acknowledgement

This paper is a result of the project supported by the grants No. 1ET100300517 and MSM0021620838. A preliminary version of this contribution was presented at the conference

Formal Description of Slavic Languages 6.5, December 2006 (to appear in the Proceedings of FDSL 6.5).

Bibliography

[Hajič2005] Hajič, Jan. 2005. Complex Corpus Annotation: The Prague Dependency Treebank. In Mária Šimková, editor, *Insight into Slovak and Czech Corpus Linguistics*. Veda Bratislava, Slovakia, pages 54–73.

[Hajičová, Partee, and Sgall1998] Hajičová, Eva, Barbara Partee, and Petr Sgall. 1998. *Topic-Focus Articulation, Tripartite Structures, and Semantic Content.* Kluwer, Dordrecht.

[Hajičová2006] Hajičová, Eva. 2006. K některým otázkám závislostní gramatiky. *Slovo a slovesnost*, (67):3–26.

[Havelka2005] Havelka, Jiří. 2005. Projectivity in Totally Ordered Rooted Trees: An Alternative Definition of Projectivity and Optimal Algorithms for Detecting Non-Projective Edges and Projectivizing Totally Ordered Rooted Trees. *The Prague Bulletin of Mathematical Linguistics*, (84):13–30.

[Holan et al.2000] Holan, Tomáš, Vladislav Kuboň, Karel Oliva, and Martin Plátek. 2000. On Complexity of Word Order. *Les grammaires de dépendance – Traitement automatique des langues*, 41(1):273–300.

[Lopatková, Plátek, and Kuboň2005] Lopatková, Markéta, Martin Plátek, and Vladislav Kuboň. 2005. Modeling Syntax of Free Word-Order Languages: Dependency Analysis by Reduction. In *Lecture Notes in Artificial Intelligence, Proceedings of Text, Speech and Dialogue*, volume 3658, pages 140–147, Heidelberg. Springer Verlag.

[Messerschmidt et al.2006] Messerschmidt, Hartmut, František Mráz, Friedrich Otto, and Martin Plátek. 2006. Correctness Preservation and Complexity of Simple RL-Automata. In *Lecture Notes in Computer Science*, volume 4094, pages 162–172, Heidelberg. Springer Verlag.

[Mikulová et al.2006] Mikulová, Marie, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, and Zdeněk Žabokrtský. 2006. Annotation on the Tectogrammatical Level in the Prague Dependency Treebank. Annotation Manual. Technical Report 30, ÚFAL MFF UK, Prague. trans. Součková, K., Böhmová, A., Čermáková, K., Havelka, J., Corness, P.

[Otto2006] Otto, Friedrich. 2006. Restarting Automata. In Z. Ésik, C. Martin-Vide, and V. Mitrana, editors, *Recent Advances in Formal Languages and Applications, Studies in Computational Intelligence*, volume 25, pages 269–303, Berlin. Springer.

[Panevová1974] Panevová, Jarmila. 1974. On Verbal Frames in Functional Generative Description. *The Prague Bulletin of Mathematical Linguistics*, (22):3–40.

[Petkevič1995] Petkevič, Vladimír. 1995. A New Formal Specification of Underlying Structure. *Theoretical Linguistics*, 21(1).

[Plátek1982] Plátek, Martin. 1982. Composition of Translation with D-trees. In *Proceedings of COLING'82*, pages 313–318.

[Plátek and Sgall1978] Plátek, Martin and Petr Sgall. 1978. A Scale of Context-Sensitive Languages: Applications to Natural Language. *Information and Control*, 38(1):1–20.

[Sgall, Hajičová, and Panevová1986] Sgall, Petr, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects.* Reidel Publishing Company, Dordrecht. ed. Mey, J.

[Sgall et al.1969] Sgall, Petr, Ladislav Nebeský, Alla Goralčíková, and Eva Hajičová. 1969. *A Functional Approach to Syntax in Generative Description of Language.* American Elsevier Publishing Company, Inc., New York.

[Šmilauer1966] Šmilauer, Vladimír. 1966. *Novočeská skladba.* SPN, Praha. 2*nd* edition.