ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2007, Vol.36, No.2

A SYSTEM FOR AUTOMATIC SYNTACTIC ANALYSIS OF LITHUANIAN SIMPLE SENTENCES

Daiva Šveikauskienė

Department of Recognition Processes, Institute of Mathematics and Informatics Vilnius
Goštauto St. 12-210, LT-01108 Vilnius, Lithuania

Abstract. The problem of automation of syntactic analysis of the Lithuanian simple sentences is investigated. The features of Lithuanian language – great inflexion and free word order in a sentence – which raise the specific claims for solving the problem of syntactic analysis of Lithuanian sentences are highlighted. The formalized procedure of syntactic analysis is based on the Backus and Naur formalism. The possibilities of extension of the boundaries of the procedure for syntactic analysis are shown. The article presents an algorithm and software for syntactic analysis of Lithuanian simple sentences. The accuracy of performance of the system was evaluated.

Keywords: artificial intelligence, computer-linguistic, natural language processing, automatic syntactic analysis.

1. Introduction

Automatic syntactic analysis belongs to the area of artificial intelligence. Its main application is machine translation. The syntactic structure of the same sentence can differ in various languages and during translation the syntactic structure of a sentence in one language gets changed into the syntactic structure of the sentence in the other language. That is why one must have computer-made syntactic structure of the sentence. Other area of application of automatic syntactic analysis is word processing systems, which include spelling checkers and by performing a syntactic analysis of sentences can point out possible errors [8].

2. Statement of the problem

Three main strategies are used for syntactic analysis – statistical approach, method on the ground of algorithm and method based on the formal description of syntax. It is decided to use the last mentioned method for the syntactic analysis of Lithuanian sentences, because the free word order in Lithuanian language causes many problems for statistical approach and by using algorithm method the amount of programming task increases very much – most every word should be processed by a separate algorithm. This statement can be illustrated by an example taken from the system of syntactic analysis for Russian language, which was created in Kiev. Looking-for dependent words of preposition *между* (between) could be taken as an instance of algorithm-based syntactic analysis. Particular algorithm called 'search of dependent word' finds the first (dependent on this preposition) word, very control algorithm looks for the second dependent on this preposition word, by examining each word in instrumentalis case in accordance with three hypothesis:

1. Is this noun dependent on the verb immediate to the left (*<u>управлять</u> программой* – to control the program)?
2. Enters this noun into a nominal phrase and depends on one word of this phrase (*организация <u>вызова</u> нужного алгоритма управляющей <u>программой</u>* – organization of <u>call</u> to the needed algorithm <u>by</u> control <u>program</u>)?
3. Is the noun dependent of preposition *между* (between) in the context to the left?

Only the words previous to current are checked according to the third hypothesis [12].

This example shows that processing of each word is specific. Such a method was taken for the Russian language because the formal description of this language, whose syntax distinguishes by great complexity [13], is very difficult. The first attempt to describe the syntax of Russian language in BNF (Backus and Naur form) in 1978 was unsuccessful, i.e. in [11] three members are distinguished in the level of parts of sentence. In BNF they are described as follows:

<subject> ::= <nominal phrase> |
    <nominal word combination> |
    <prepositional phrase> |
    <prepositional word combination>;

<predicate> ::= <verbal phrase> |
<verbal word combination>;

<complementary adverbial construction> ::=
(<nominal phrase> |
<nominal word combination> |
<prepositional phrase> |
<prepositional word combination>;
if they are not <subject>) |
(<infinitive> | <several infinitives>;
if they don't enter into predicate) |
(<adverb> | <several adverbs>;
if they are not attributes of the <subject>)

It is in evidence that the given description of part of sentence don't go into the framework of BNF. The underlined positions point out where the description contravenes the rules of BNF. Thus in the early systems it failed to describe the syntax of Russian language by context-free grammar, as it was done for English language.

In the new Internet publications, Syntax of Russian language is given as rules of descriptive character. With the help of these rules, the syntactic groups are formed. These syntactic groups are the aim of syntactic analysis. The group consists of information about what is the first and the last word, what is a type of group (i.e. adverb-adjective), what is the main word or subgroup. The group also includes the morphologic data necessary for its relations with other words out of group. As an example of the rule, the description of group 'adjective noun' could be given.

The rule for formation of nominal group (ADJECTIVE-NOUN)

Object: String: several groups, whose main word is adjective – group, whose main word is noun.

Condition: The main word of all the groups must agree with the main word of nominal group by gender, number and case.

The main group: nominal group

Type: nominal group agreed by gender, number and case.

Examples: длинная унылая дорога (a long monotonous way); единственному настоящему другу (to the only true friend).

Another method, which we decline, is a statistical approach.

An attempt to make the syntactic analysis of Russian language by using statistical data is described in [14]. It was proposed to form the syntactic structure of a sentence by using statistical processing of texts as opposed to linguistic research. The researchers stated that any linguistic unit (word, grammatical category, syntactic construction) could appear in text with certain probability. For instance, the frequentative valence dictionary of English language shows that 195 distinct patterns of syntactic relations are typical for verb and 10 common patterns make 86% of all cases.

Statistical regularity is applied to the linear structure of sentence. Any class of words has its own frequency of position against the beginning of the sentence. English verb is in 2-5 position with probability 0.7 and in 2-10 position with the probability 0.95 [14]. On the ground of these data an algorithm for search of syntactic relations of verb is designed. It operates under a principle of presumption. In certain positions of a sentence (at the beginning of a sentence, at the position of the predicate and at the positions of words subordinated to it), presumptions are made about eventual results considering the type of a verb and its environment, the length of sentence and so on. Presumptive events are situated in decreasing order. If the presumptions in the analysis points coincide, a group of words related to a verb is singled out. Then relations between the group of verb and others words of sentence are established [14].

It seems not very likely that syntactic analysis based on statistical arrangement of parts of sentence could give good results for Lithuanian language. Due to free word order in a sentence the probabilities that the word should be at the beginning in the middle and at the end of sentence differ not very much in contrast to English language, where the very grammar rules demand the second position for verb. Consequently, the probability of verb in the 2-5th position in a sentence is rather big. In the Lithuanian language the change of word order is used as synonymous variant on purpose to avoid the monotony, thus it is little possible to gather from a position of the word about its syntactic function. In Internet we didn't find recent publications concerning the statistic syntactic analysis.

Considering the weakness of both mentioned methods – stochastic approach and method of algorithm – the strategy based on the formal description of grammar rules, which was used for English systems of automatic syntactic analysis was taken for the automatic syntactic analysis of Lithuanian language. If the to solving problem is described by context free grammar (BNF – Backus and Naur form) the programming is simply.

The common strategy of parsing of English sentences could be illustrated by an example of the sentence *The green water evaporated* [1], where the word green may be an adjective or a noun, the word water may be a noun or verb and rewriting rules of context-free grammar are presented in Figure 1.

Every step of analysis is performed considering the information at hand. The parsing begins in every step by processing a leftmost symbol. If it is an English word, i.e. terminal symbol, the next symbol is checked, otherwise the grammar is used to rewrite the first symbol.

Initially sentence S is replaced by noun phrase NP and verb phrase VP using Rule 1 and it is checked whether they succeed: at first leftmost symbol NP is processed and it is checked according to Rule 2 whether an article and a noun are at the beginning of the sentence. Such word combination is possible (*the* is an article according to Rule 7 and *green* is a noun according to Rule 8), thus it is stored. Further the first alternative of symbol VP is checked (Rule 4). It succeeds, i.e. the third word in the sentence may be a verb according to Rule 11. Thus, the sentence structure is produced and it is presented in Figure 2.

S –> NP VP	(Rule 1)		ART –> the	(Rule 7)
NP –> ART NOUN	(Rule 2)		NOUN –> green	(Rule 8)
NP –> ART ADJ NOUN	(Rule 3)		NOUN –> water	(Rule 9)
VP –> V	(Rule 4)		ADJ –> green	(Rule 10)
VP –> AUX V	(Rule 5)		V –> water	(Rule 11)
VP –> V NP	(Rule 6)	its	V –> evaporated	(Rule 12)
			AUX –> can	(Rule 13)

Figure 1. Context free grammar for the parsing of sentence *The green water evaporated*

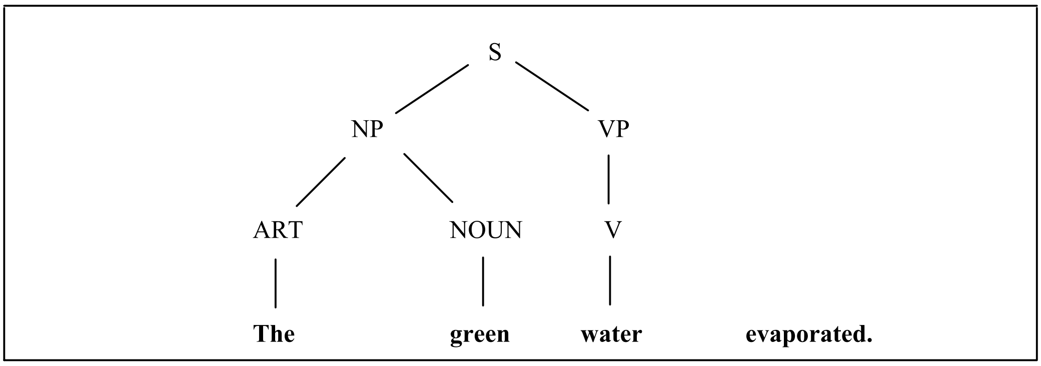


Figure 2. Syntactic structure of the sentence *The green water*

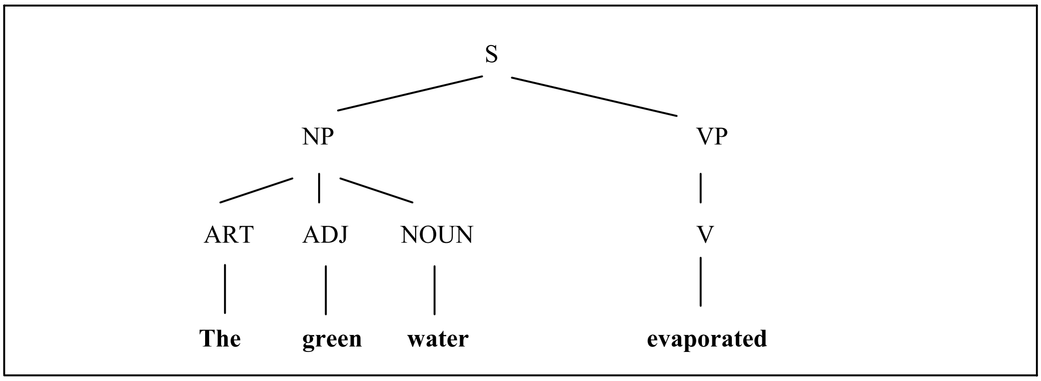


Figure 3. Syntactic structure of the sentence *The green water evaporated*

But this structure cannot account for the word evaporated, thus it is rejected. Then it is returned to the second alternative of VP (Rule 5). Since the word *water* cannot be auxiliary, the program returns to the second alternative of NP (Rule 3). It succeeds: *the* may be an article according to Rule 7, *green* may be an adjective according to Rule 10 and *water* may be a noun according to Rule 9. Thus the first alternative of VP (Rule 4) is checked again. It succeeds: *evaporated* can be a verb according to Rule 12. In this instance all words of the sentence are processed. This structure is presented as a result of analysis (Figure 3).

In principle, the parsing of an English sentence is the finding of bound, where the subject ends and the predicate begins. They are arranged strongly in sequence in the English language. In the Lithuanian language with free word order we cannot avail of it.

Two features of Lithuanian language could be regarded as typical: the great inflexion (many word forms with different endings) and free word order in a sentence. Therefore in Lithuanian sentences namely endings are determinative when deciding which syntactic function the word performs.

In the English language the syntactic structure of a sentence is established regarding the word order in a sentence, which in principle determines the syntactic function of a word. These propositions could be illustrated by the following example – two sentences in the English and Lithuanian languages:

Vaikas valgo obuolį. – *The child eats the apple.*

Obuolį valgo vaikas. – **The apple eats the child.*

In the Lithuanian sentences the ending *-į* of the word *obuolį* shows that it is accusative and consequently this word is treated as an object, because the verb (*valgyti – to eat*) in this sentence governs the accusative case without reference to its position in a sentence. In the second English sentence the word *apple* is already a subject and this syntactic function for it is determined by the position in a sentence.

Thus, one may conclude that syntactic information acquisition from word flexions is not envisaged in automatic syntactic systems of the English language. Therefore it was necessary to develop a peculiar method to identify the parts of sentence in the Lithuanian language.

3. Solving of the problem

3.1. Method of the formalization of syntactic analysis

Lithuanian sentences should be analysed by taking into account the specific features of the Lithuanian language – great inflexion and free word order in a sentence. The Lithuanian language has no strict, grammatically defined word order [4], so it is absolutely impossible to rely on the position of a word in a sentence, and it is necessary to estimate a lot of flexions that bear the main syntactic information load.

When creating the formal grammar to describe the Lithuanian syntax, all the syntactic functions are differentiated according to morphological categories of the words expressing them, so as to find words connected by a direct syntactic relation in accordance with morphological categories.

The gist of the Lithuanian syntactic analysis method is that, when creating the formal grammar in BNF (Backus and Naur form), all syntactic functions are differentiated according to the parts of speech by which they can be expressed, for example the subject is expanded into the subject expressed by a noun, the subject expressed by a pronoun, the subject expressed by infinitive of a verb:

```
<SUBJECT>::= <SUBJECT-NOUN>|
             <SUBJECT-PRONOUN>|
             <SUBJECT-INFINITIVE>;
```

Further, we continue decomposing according to morphological categories typical of each part of speech, for example the first alternative of subject (the subject expressed by a noun) is divided according to case, number and gender:

```
<SUBJECT-NOUN>::=
    <SUBJECT-NOUN-NOMINATIVE-SINGULAR-
MASCULINE>|<SUBJECT-NOUN-NOMINATIVE-SINGULAR-
FEMININE>    |<SUBJECT-NOUN-NOMINATIVE-PLURAL-
MASCULINE>   |<SUBJECT-NOUN-NOMINATIVE-PLURAL-
FEMININE>;
```

Next, according to the congruence of morphological categories we look for directly dependent words. The syntactic relation between the object expressed by a noun in dative case, singular of feminine gender and its attribute, expressed by adjective in dative case, singular of feminine gender, is treated as a thread between these parts of speech.

```
<THREAD#OBJECT-NOUN-DATIVE-SNGULAR-FEMININE+
    ATTRIBUTE-ADJECTIVE-DATIVE-SNGULAR-
FEMININE>
```

Regarding the free word order in Lithuanian sentences, BNF rules are composed for finding direct syntactic relations, by indicating which words can be between two words related by a direct syntactic link. The information about the words, which can intervene into the word combination, is described by a non-terminal symbol INSERTION.

```
<THREAD#OBJECT-NOUN-DATIVE-SNG-FEM+
    AGREEING-ATTRIBUTE-ADJECTIVE-DATIVE-SNG-
FEM> ::=
    <AGREEING-ATTRIBUTE-ADJECTIVE-DATIVE-SNG-
FEM>
    <INSERTION#OBJECT-NOUN-DATIVE-SNG-FEM+
    ATTRIBUTE-ADJECTIVE-DATIVE-SNG-FEM>
    <OBJECT-NOUN-DATIVE-SNG-REM>;
```

These issues are described in more details in [6].

3.2. Algorithm

An algorithm for syntactic analysis of simple Lithuanian sentences is presented in Figure 4. When making a syntactic analysis, a sentence given by a user serves as initial data. At first, it is decomposed into words. Next, the morphological form of each word is defined. Then, the syntactic functions of words are defined according to the morphological categories. Further, the direct syntactic relations between the words are pursued by the congruence of morphological categories. If the syntactic relationships are established for all the words of a sentence (i.e. there is no word without a relation with another word), a graphical image of the syntactic structure of the sentence is built and displayed. If at least one word has been left without any relation to another word, such a sentence is regarded as impossible.

The running of the algorithm of syntactic analysis can be illustrated by the sentence *Tamsūs pušų sakai blizgėjo saulėje* (Dark resin of pine trees was glittering in the sun). Figure 5 illustrates the operation results of blocks of segmentation and morphologic analysis by an example. Due to morphological polysemy two variants of word *sakai* (resin) and *blizgėjo* (was glittering) are given.

It should be taken into account that any given morphological form of a word can be true in the sentence. Therefore it is indispensable to look through all the possible combinations, i.e. to make as many sentences with the information about morphological categories of words as there are different possible variants of a sentence. There are four of them in this example, presented in Figure 6. This is in essence the same sentence, only the words here are replaced by their morphological categories. The variants obtained could be called as morphological sentences.

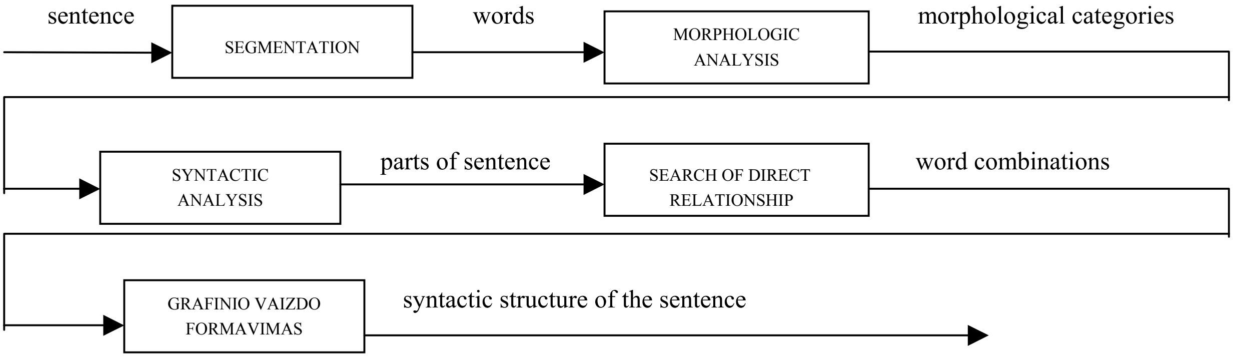


Figure 4. Block-scheme of the algorithm

SENTENCE	Tamsūs pušų sakai blizgėjo saulėje. (Dark resin of pine trees was glittering in the sun)				
SEGMENTS	Tamsūs (dark)	pušų (of pine trees)	sakai (resin)	blizgėjo (was glittering)	saulėje (in the sun)
MORPHOLOGIC	Būdv-vard-dgs-vyrg (Adjective-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Daikt-vard-dgs-vyrg (noun-nominative-plural-masculine)	Veiksm-vns-3a (verb-singular-3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)
			Veiksm-vns-2a Verb-singular-2person)	Veiksm-dgs-3a (verb-plural-3person)	

Figure 5. Illustration of segmentation and morphologic analysis by using an example

1.	Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Daikt-vard-dgs-vyrg (noun-nominative-plural-masculine)	Veiksm-vns-3a (verb-singular-3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)
2.	Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Veiksm-vns-2a Verb-singular-2person)	Veiksm-vns-3a (verb-singular-3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)
3.	Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Daikt-vard-dgs-vyrg (noun-nominative-plural-masculine)	Veiksm-dgs-3a (verb-plural-3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)
4.	Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	V-eiksm-vns-2a Verb-singular-2person)	Veiksm-dgs-3a (verb-plural-3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)

Figure 6. Variants of morphological sentences

SYNTACTIC ANALYSIS	DERIN-PAŽYM (AGREEING ATRIBUTE)	NEDERIN-PAŽYM (NONAGREEING-ATRIBUTE)	VEIKSN (SUBJECT)	TARIN (PREDICATE)	APLINKYB (ADVERBIAL-MODIFIER)
	Būdv-vard-dgs-vyrg	Daikt-kilm-dgs-motg	Daikt-vard-dgs-vyrg	Veiksm-vns-3a	Daikt-viet-vns-motg
		PAPILD (OBJECT)			
		Daikt-kilm-dgs-motg			

Figure 7. Illustration of the syntactic analysis by an example

1.1

DERIN-PAŽYM (AGREEING ATTRIBUTE)	**NEDERIN-PAŽYM (NONAGREEING ATTRIBUTE)**	VEDIKSN (SUBJECT)	TARIN (PREDICATE)	APLINKYB (ADVERBIAL MODIFIER)
Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Daikt-vard-dgs-vyrg **noun**-nominative-plural-masculine	Veiksm-vns-3a (verb-**singular-**3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)

1.2

DERIN-PAŽYM (AGREEING ATTRIBUTE)	**PAPILD (OBJECT)**	VEDIKSN (SUBJECT)	TARIN (PREDICATE)	APLINKYB (ADVERBIAL MODIFIER)
Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Daikt-vard-dgs-vyrg **noun**-nominative-plural-masculine	Veiksm-vns-3a (verb-**singular-**3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)

2.1

DERIN-PAŽYM (AGREEING ATTRIBUTE)	**NEDERIN-PAŽYM (NONAGREEING ATTRIBUTE)**	TARIN (PREDICATE)	TARIN (PREDICATE)	APLINKYB (ADVERBIAL MODIFIER)
Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Veiksm-vns-2a **Verb**-singular-**2person**)	Veiksm-vns-3a (verb-**singular-**3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)

2.2

DERIN-PAŽYM (AGREEING ATTRIBUTE)	**PAILD (OBJECT)**	TARIN (PREDICATE)	TARIN (PREDICATE)	APLINKYB (ADVERBIAL MODIFIER)
Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Veiksm-vns-2a **Verb**-singular-**2person**)	Veiksm-vns-3a (verb-**singular-**3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)

3.1

DERIN-PAŽYM (AGREEING ATTRIBUTE)	**NEDERIN-PAŽYM (NONAGREEING ATTRIBUTE)**	VEDIKSN (SUBJECT)	TARIN (PREDICATE)	APLINKYB (ADVERBIAL MODIFIER)
Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Daikt-vard-dgs-vyrg **noun**-nominative-plural-masculine	Veiksm-dgs-3a (verb-**plural-**3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)

3.2

DERIN-PAŽYM (AGREEING ATTRIBUTE)	**PAILD (OBJECT)**	VEDIKSN (SUBJECT)	TARIN (PREDICATE)	APLINKYB (ADVERBIAL MODIFIER)
Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Daikt-vard-dgs-vyrg **noun**-nominative-plural-masculine	Veiksm-vns-3a (verb-**plural-**3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)

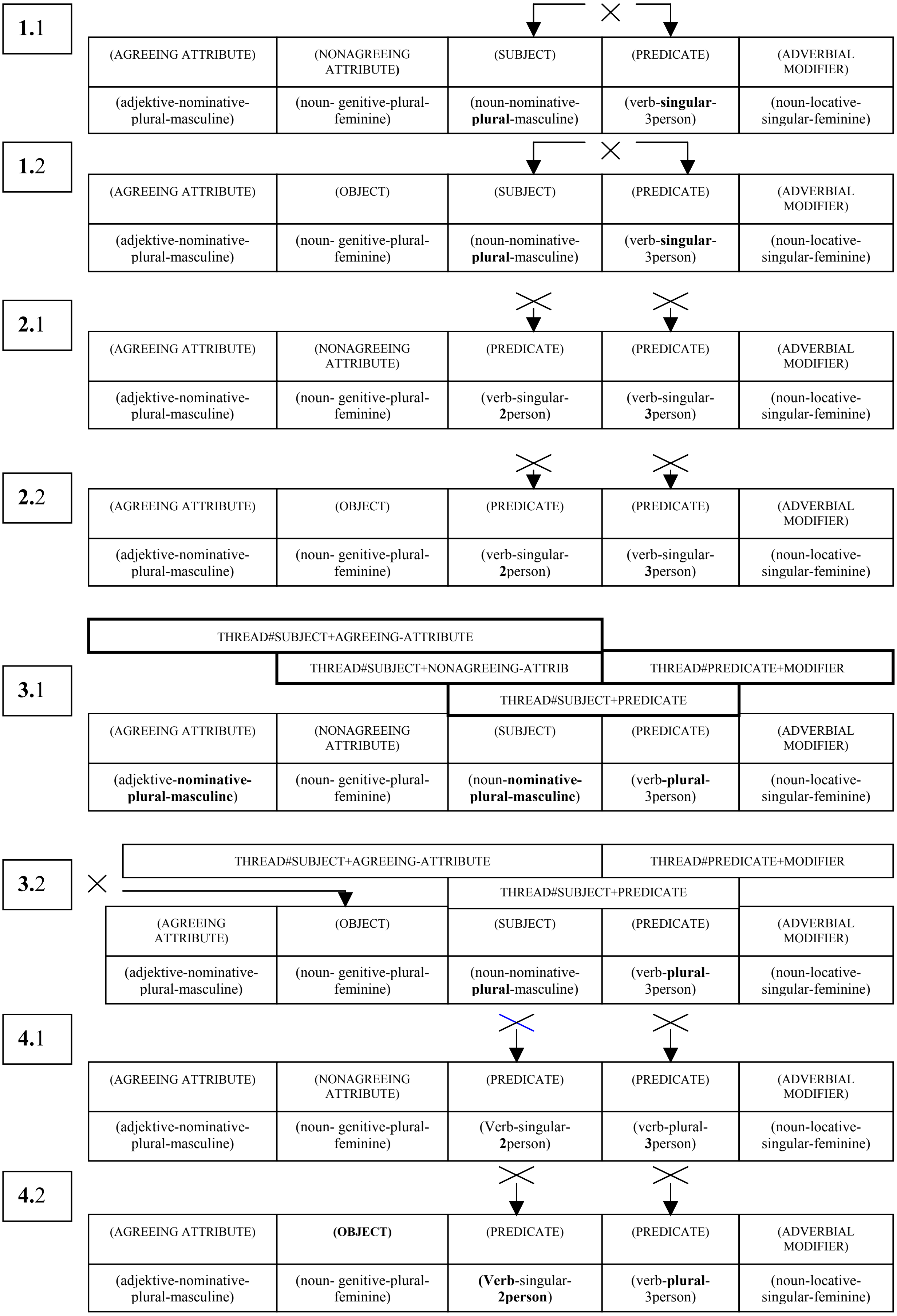
4.1

DERIN-PAŽYM (AGREEING ATTRIBUTE)	**NEDERIN-PAŽYM (NONAGREEING ATTRIBUTE)**	TARIN (PREDICATE)	TARIN (PREDICATE)	APLINKYB (ADVERBIAL MODIFIER)
Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Veiksm-vns-2a **Verb**-singular-**2person**)	Veiksm-vns-3a (verb-**plural-**3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)

4.2

DERIN-PAŽYM (AGREEING ATTRIBUTE)	**PAILD (OBJECT)**	TARIN (PREDICATE)	TARIN (PREDICATE)	APLINKYB (ADVERBIAL MODIFIER)
Būdv-vard-dgs-vyrg (adjektive-nominative-plural-masculine)	Daikt-kilm-dgs-motg (noun- genitive-plural-feminine)	Veiksm-vns-2a **Verb**-singular-**2person**)	Veiksm-vns-3a (verb-**plural-**3person)	Daikt-viet-vns-motg (noun-locative-singular-feminine)

Figure 8. Variants of syntactic sentences



1.1

(AGREEING ATTRIBUTE)	(NONAGREEING ATTRIBUTE**)**	(SUBJECT)	(PREDICATE)	(ADVERBIAL MODIFIER)
(adjektive-nominative-plural-masculine)	(noun- genitive-plural-feminine)	(noun-nominative-**plural**-masculine)	(verb-**singular**-3person)	(noun-locative-singular-feminine)

1.2

(AGREEING ATTRIBUTE)	(OBJECT)	(SUBJECT)	(PREDICATE)	(ADVERBIAL MODIFIER)
(adjektive-nominative-plural-masculine)	(noun- genitive-plural-feminine)	(noun-nominative-**plural**-masculine)	(verb-**singular**-3person)	(noun-locative-singular-feminine)

2.1

(AGREEING ATTRIBUTE)	(NONAGREEING ATTRIBUTE)	(PREDICATE)	(PREDICATE)	(ADVERBIAL MODIFIER)
(adjektive-nominative-plural-masculine)	(noun- genitive-plural-feminine)	(verb-singular-**2**person)	(verb-singular-**3**person)	(noun-locative-singular-feminine)

2.2

(AGREEING ATTRIBUTE)	(OBJECT)	(PREDICATE)	(PREDICATE)	(ADVERBIAL MODIFIER)
(adjektive-nominative-plural-masculine)	(noun- genitive-plural-feminine)	(verb-singular-**2**person)	(verb-singular-**3**person)	(noun-locative-singular-feminine)

3.1

THREAD#SUBJECT+AGREEING-ATTRIBUTE

THREAD#SUBJECT+NONAGREEING-ATTRIB

THREAD#PREDICATE+MODIFIER

THREAD#SUBJECT+PREDICATE

(AGREEING ATTRIBUTE)	(NONAGREEING ATTRIBUTE)	(SUBJECT)	(PREDICATE)	(ADVERBIAL MODIFIER)
(adjektive-**nominative-plural-masculine**)	(noun- genitive-plural-feminine)	(noun-**nominative-plural-masculine**)	(verb-**plural**-3person)	(noun-locative-singular-feminine)

3.2

THREAD#SUBJECT+AGREEING-ATTRIBUTE

THREAD#PREDICATE+MODIFIER

THREAD#SUBJECT+PREDICATE

(AGREEING ATTRIBUTE)	(OBJECT)	(SUBJECT)	(PREDICATE)	(ADVERBIAL MODIFIER)
(adjektive-nominative-plural-masculine)	(noun- genitive-plural-feminine)	(noun-nominative-**plural**-masculine)	(verb-**plural**-3person)	(noun-locative-singular-feminine)

4.1

(AGREEING ATTRIBUTE)	(NONAGREEING ATTRIBUTE)	(PREDICATE)	(PREDICATE)	(ADVERBIAL MODIFIER)
(adjektive-nominative-plural-masculine)	(noun- genitive-plural-feminine)	(Verb-singular-**2**person)	(verb-plural-**3**person)	(noun-locative-singular-feminine)

4.2

(AGREEING ATTRIBUTE)	**(OBJECT)**	(PREDICATE)	(PREDICATE)	(ADVERBIAL MODIFIER)
(adjektive-nominative-plural-masculine)	(noun- genitive-plural-feminine)	**(Verb**-singular-**2person)**	(verb-**plural**-3person)	(noun-locative-singular-feminine)

Figure 9. Establishment of direct syntactic relations

The execution of the block of syntactic analysis (Figure 4) is shown in more detail in Figure 7. The word *pušų* (pine trees) can perform two syntactic functions: that of an object and an attribute. It should be regarded here that any of them could be true in the given sentence. Therefore, the eight possible variants of the sentence are given. All of them are illustrated in Figure 8. A sentence, in which the words are replaced by their syntactic functions, is called a syntactic sentence in this work.

The block of search of direct relationships is shown in more detail in Figure 9. In search of syntactic relations according to the congruence of morphological categories, all the variants that do not belong to the given sentence are rejected. The example (Figure 9) demonstrates that out of the eight variants only one is treated as possible. Combinations 1.1 and 1.2 are rejected due to non-congruence of morphological categories: the subject is expressed by a nominative case of a noun, plural, while the predicate by a singular form of a verb. In combinations 2.1 and 2.2 as well as in 4.1 and 4.2 subject for predicates cannot be found, i.e. the nucleus of the sentence is not established. In combination 3.2, one word remains without any relations to other words therefore this variant is also treated as an impossible sentence. Thus the combination 3.1 only is regarded as true variant, which coincides with the given sentence.

When analysing another sentence, the amount of morphologic and syntactic variants may differ, consequently, the amount of both the morphologic and syntactic sentences. The final variant, however, would be the same: only one possible variant out of all the obtained ones is treated true.

The block of search of direct relationships is shown in more detail in Figure 9. In search of syntactic relations according to the congruence of morphological categories, all the variants that do not belong to the given sentence are rejected. The example (Figure 9) demonstrates that out of the eight variants only one is treated as possible. Combinations 1.1 and 1.2 are rejected due to non-congruence of morphological categories: the subject is expressed by a nominative case of a noun, plural, while the predicate by a singular form of a verb. In combinations 2.1 and 2.2 as well as in 4.1 and 4.2 subject for predicates cannot be found, i.e. the nucleus of the sentence is not established. In combination 3.2, one word remains without any relations to other words therefore this variant is also treated as an impossible sentence. Thus the combination 3.1 only is regarded as true variant, which coincides with the given sentence.

When analysing another sentence, the amount of morphologic and syntactic variants may differ, consequently, the amount of both the morphologic and syntactic sentences. The final variant, however, would be the same: only one possible variant out of all the obtained ones is treated true.

3.3. Software

The program for syntactic analysis of the Lithuanian sentences is written in the Visual Basic'6 programming language on purpose to simply data interception from the morphological analysis program that has already been written in this language, because the results of morphological analysis are used for the syntactic analysis. The block-scheme of software for Lithuanian syntactic analysis is presented in Figure 10.

By performing the syntactic analysis the input data consist of a sentence, which is inputted from the text field. It is assigned to the variable 'sentence'. By function 'Trim' all the gaps are deleted in front of the first word and behind the end of the sentence. At first, the sentence should be decomposed into words. This is the job of subroutine SEGMENTATION. Data are transmitted through three parameters. The sentence is transferred to the subroutine as a string-type variable and data of two types are returned: a variable 'number of words', which contains the information how many words makes up the sentence, and one-dimensional array 'segments' each element of which corresponds to one word in the sentence. The array only includes those words that can perform a syntactic function (conjunctions, particles and other words that cannot be a pat of sentence, are rejected, however information on their place in the sentence is preserved in a separate file so it can be used later if needed). The elements of array are numbered starting with 1 and their index corresponds to a word's place in the sentence, i.e. in the array the words are arranged in the same order as they were in the sentence.

Next, the subroutine MORPHOLOGY is called by submitting to it the number of words via parameters as well as the sentence decomposed into words, i.e. the array 'segments'. In this work the software for Lithuanian morphologic analysis is used, which is created by Vytautas Zinkevičius (more details in [10]). It is accessible as a file of the Dynamic Link Library (DLL).

The Lithuanian language has not so sharp polysemy problem as the English language where almost every verb is concurrently a noun. Nevertheless, sometimes there may appear several morphological meanings of the same word in the Lithuanian language as well. During the morphologic analysis several possible variants are sometimes obtained. 47% of Lithuanian words are polysemous [5]. Most frequently, one can face coincidental forms of nouns of feminine gender, singular in nominative and instrumental cases as well as that of singular nominative and plural genitive cases etc. Therefore the subroutine of morphological analysis returns not a one-dimensional array, but a three-dimensional one (x, y, z). Ten possible meanings are envisaged for each word, i.e. the number of array elements in the z-direction is 10; in the y-direction, words of the sentence are presented (i.e. the element number shows the current position of a word in the sentence) and in the x-direction 7 elements, containing information on a word in morphological, syntactic and semantic aspects are arranged. Their signification is given in the table 1.

Table 1. Signification of 7 elements in the x-direction of the array of grammatical data

Current index number (in x-direction)	Nature of the stored information
1	Part of speech
2	Case
3	Number
4	Gender
5	Person
6	Semantic features
7	Syntactic function

If the part of speech has not got some morphological category (e.g. a verb is not inflected by cases and a noun by person), empty string is written in the respective element, and, during processing, the program does not access this element.

The morphological subroutine writes the morphological and semantic data into the first six elements. The seventh element indicates a syntactic function and is filled by a syntactic subroutine. Information for the first five elements is extracted from the DLL (Dynamic Link Library) file by using the function of morphological analysis. Access to the morphological analysis function of this library enables to extract the grammatical information about one word. During the access to DLL it is necessary to submit three parameters to this function:

- a symbol string (a word to be analysed grammatically),
- memory address where the result is to be written.
- Area of memory size meant for the result (in order to guarantee that, during the writing of results, DLL function does not exceed the memory size reserved for it).

The morphological analysis function provides information about a word by grammatical categories. For example, the symbol string *namas* (house) is defined as follows: noun, masculine gender, singular, nominative case. The way, how it is done (which data are employed that reflect lexical and grammatical knowledge of the Lithuanian language etc.) is described in more detail in [10]. If a word corresponds to several forms, the function indicates all of them, e.g. for the symbol string *mes* it is given:

1. pronoun (we), plural, nominative;
2 verb (throw), non reflexive, direct mood, future tense, singular, 3rd person;
3. verb (throw), non reflexive, direct mood, future tense, plural, 3rd person.

In the case of a successful accomplishment of the function (i.e. if the symbol string is recognized as a Lithuanian word) the information on one or several possible forms of the word is written into the area of the result. The data on one form consist of a lemma, (the initial form of the word or entry form in vocabulary is the infinitive for verbs or the nominative case for nouns, adjectives, numerals etc.) and a code of 13 bits that contains information on a part of speech, reflexivity, voice (of participles), mood, tense, gender, number, case, person etc. Not all this information is used for syntactic analysis, because some of the data do not influence the definition of a syntactic function, e.g. voice of participles, tense of verbs and the like.

Next, semantic information is added to the morphological data obtained. Semantic data are stored in a separate file. That is a text file the records of which are read line by line. Each line contains the lemma and next the semantic features of the word separated by a gap. In this work the following semantic features are used:

- The feature of time for nouns; this feature is attached to such words as week, day, month, autumn and so on;
- The feature of place for nouns;
- The feature of quantity for nouns;
- The feature of person for pronouns, because the morphologic analysis system do not give the information of such kind;
- List of cases for verbs, which are governed by the verb; if the verb governs several cases, all they are written in a list by separating them with the help of a gap;
- List of prepositions governed by the verb by signifying them as a preposition with a case;
- For prepositions the very same word is written as its semantic feature, because the preposition can determine syntactic function of a case, e.g. preposition *už (for)* with noun in accusative case performs the syntactic function of an object, and the same preposition *už (behind)* with noun in genitive case performs the function of an adverbial modifier.

In the file of semantic features it is searched for every lemma found during the morphological analysis. The lemma is compared with the first symbols of the string. When such a record is found, the remaining part of string is copied to the semantic field of the word. These data are used later by deciding which syntactic function performs current word. The file of semantic features can be edited separate from the program. Therefore, it is always possible to add to the system a new word or a new feature. In this work the above-mentioned file only embraces that information which is used in BNF description i.e. which is necessary for the syntactic analysis. It will be observed that it is not a very comprehensive semantic description of a word because the above mentioned file embraces only those features which are necessary for the needs of syntactic analysis i.e. which may determine the syntactic function of a word or its relation to other words. For example, accusative case of noun performs usually the syntactic function of an object (*skaityti knygą – to read the book*), but if the noun has a feature of time, it performs syntactic function of adverbial modifier (*skaityti naktį – to read at night*) though the verb governs accusative. If both such accusative cases (with the feature of time and without it) are in a sentence (*Šią naktį ji skaitė knygą – She read a book this night)*, the semantic features only decide the syntactic function of a word. Semantic information is only given for nouns, pronouns and verbs, which are in first 2000 words in Lithuanian frequency dictionary. The frequency dictionary allows select the most needful words. This proposition is affirmed by rating: ten most frequently used Lithuanian words make 12.33% of texts and they make total just 0.03% of all words occurred in the texts. Fifty most frequently used words make 24.27% of texts, five hundred words make more than 50%, and two thousand words make even 75.50% of texts [3]. Thus it is reliable to get the list of most frequently used words with the help of the frequency dictionary.

The semantic data are gained from the file of semantic features. The records are arranged in strings in this file (one string is allocated for one word). The search is performed by using of the lemma (obtained during the morphological analysis) as a filter, which is compared with the symbols in the beginning of every string. The comparison is made with the help of operator '*Like'*. When finding a string the beginning of which coincides with the filter, the remaining part of the string (the semantic features) is copied to the 6th element (used for semantic features) of the array for data concerning the word. In the system the possibility is provided to extend the file of semantic features by including a new word or a new feature. This demands to write an additional string with data about the word into the file of semantic features.

When finishing the job, the subroutine MORPHOLOGY returns a three-dimensional array 'morphologic variants' which contains the information about all possible morphological forms of every word and a one-dimensional array 'number of morphologic variants' with information how many possible forms every word has. The index of array shows the position of a word in the sentence and the value of the element gives the number of morphological forms of the word.

It is necessary to check all possible combinations therefore the separate set of morphologic categories should be made for every case. This set is called in this work a morphologic sentence. It is in essence the same sentence only the morphological categories are given in it in place of words. All morphologic sentences are formed by subroutine VARIANTS. Three values (variable 'number of words', one-dimensional array 'number of morphologic variants' and three-dimensional array 'morphologic variants') are submitted through parameters to the subroutine. It forms all possible combinations of morphologic forms, i.e., makes all possible morphologic sentences and returns a three-dimensional array 'morphologic sentences'. Together the variable 'number of morphologic sentences' is returned which contains information how many morphologic sentences are made.

Then, given sentences are transferred one at a time to the subroutine SYNTAX, which establishes the syntactic function of every word according to the BNF description of Lithuanian syntax. Sometimes the same morphologic form can perform several syntactic functions: genitive case of a noun may be a non-agreeing attribute (*vilko kailis – fell of a wolf*) or an object (*bijau vilko – I am afraid of a wolf*), infinitive may be a subject (*dirbti šachtoje buvo sunku – to work in the mine was difficult*), or an object (*dainuoti ji mėgdavo – she liked to sing*), or an attribute (*noras gyventi buvo didelis – the will to live*); some prepositional constructions may be an attribute (*namas iš plytų – the house of bricks*), or an object (*išsiskirti iš bendraamžių – to distinguish from contemporaries*), or modifier (*grįžti iš miesto – to return from the town*) and so on. Accordingly, the results of the subroutine SYNTAX should be submitted to the subroutine VARIANTS, which makes the syntactic sentences analogous to the morphologic sentences. The syntactic sentences are all possible combinations of syntactic functions in the given sentence. A syntactic sentence is the same sentence in which the words are replaced by syntactic, morphologic and semantic information.

Then one tries to form the syntactic structure for every syntactic sentence. The subroutine THREADS performs it. Two parameters (variable 'number of words' and two-dimensional array 'syntactic sentence' which coincides with one combination in array of syntactic sentences) are submitted to this subroutine. First, the nucleus of the sentence is found then it is searched for the words extending the subject and the predicate according to the agreeing of morphologic categories. Herewith the semantic features are regarded. Thus the morphologic and semantic information is taken into account. In that case when the syntactic structure is successfully formed for the current syntactic sentence i.e. the threads (direct syntactic relations) are found for all the words and any word is without relation to another word, it is treated that the obtained structure is the right one for a given sentence. The information about it is returned in Boolean variable 'result of analysis'. The two-dimensional array 'syntactic relations' with information which words are related every word to and one-dimensional array 'number of syntactic relations', where the number of related words is given for every word of the sentence is returned too. If it is failed to form the syntactic structure for a current variant of syntactic sentence, one treats that such a sentence is impossible and the value of variable 'result of analysis' is returned as false. In such case, the syntactic sentence is rejected and the next combination (the next syntactic sentence is processed.

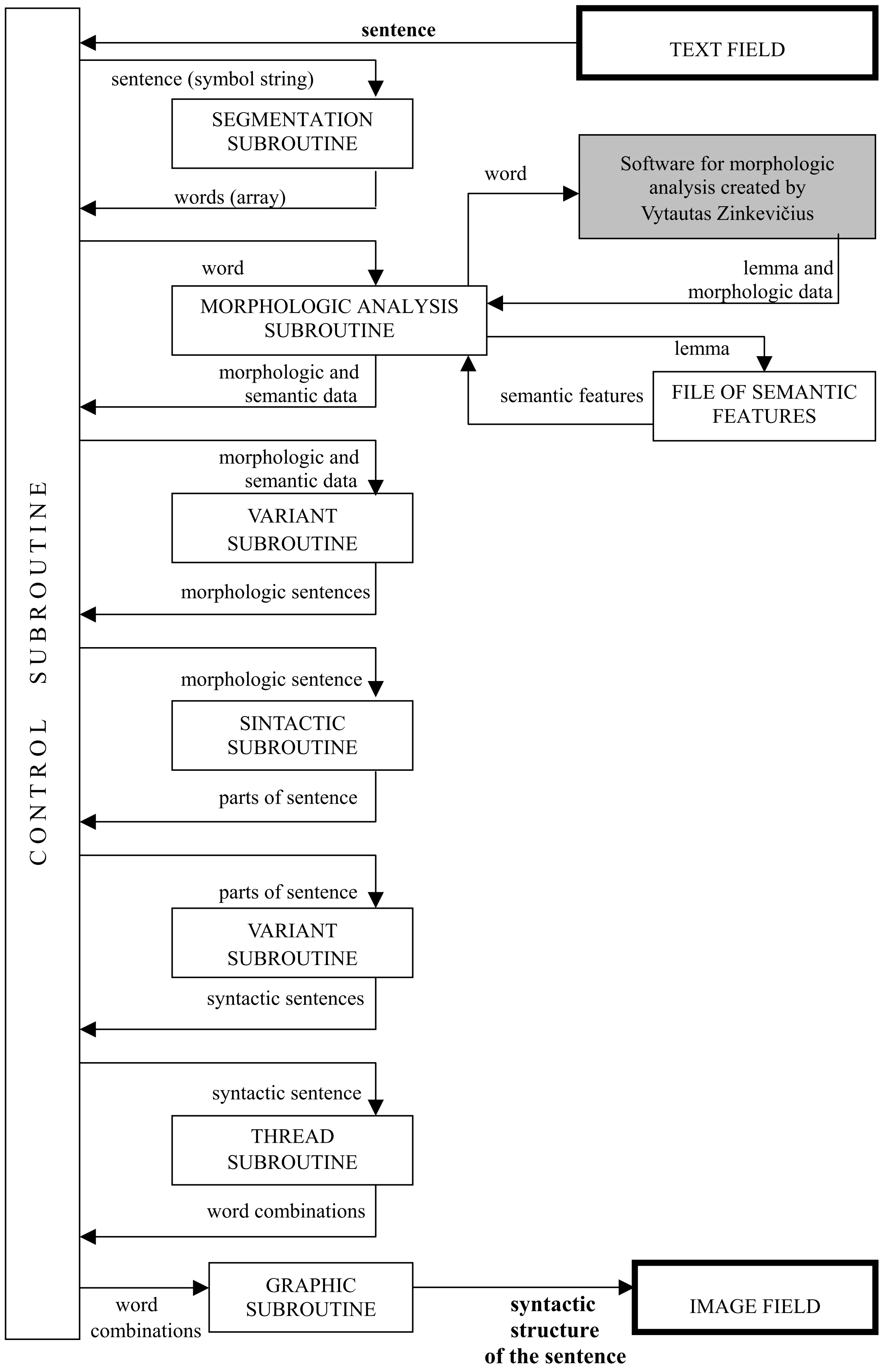
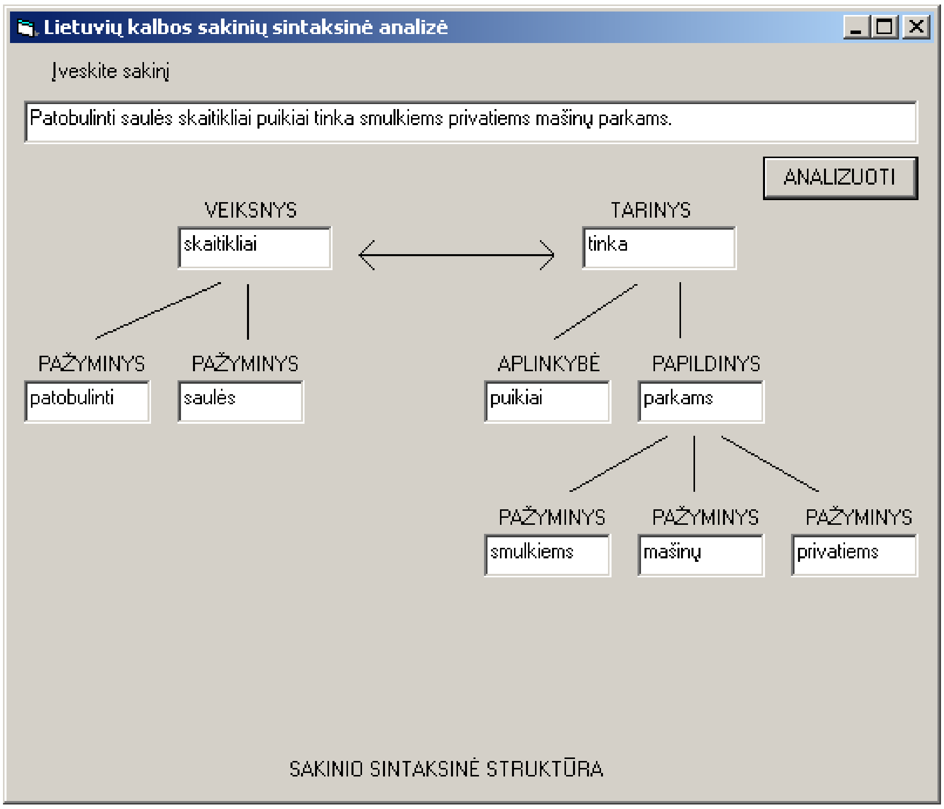


Figure 10. Block-scheme of software



Įveskite sakinį

Patobulinti saulės skaitikliai puikiai tinka smulkiems privatiems mašinų parkams.

VEIKSNYS: skaitikliai

TARINYS: tinka

PAŽYMINYS: patobulinti

PAŽYMINYS: saulės

APLINKYBĖ: puikiai

PAPILDINYS: parkams

PAŽYMINYS: smulkiems

PAŽYMINYS: mašinų

PAŽYMINYS: privatiems

SAKINIO SINTAKSINĖ STRUKTŪRA

Figure 11. Displayed syntactic structure of the sentence: *Patobulinti saulės skaitikliai puikiai tinka smulkiems privatiems mašinų parkams – Improved sun registers perfectly pass for small private car parks*

Then the parsed sentence is submitted to the subroutine GRAPHIC, which forms the view of syntactic structure of the sentence in display. If several variants of the syntactic structure for a given sentence are possible, the first obtained is displayed. Till the automatic semantic analysis is not created for the Lithuanian language, there is no possibility to decide which variant of many grammatically possible is true in the given sentence. The following parameters are submitted to the subroutine GRAPHIC: variable 'number of words', one-dimensional array 'segments', two-dimensional array 'syntactic sentence', two-dimensional array 'syntactic relations' and one-dimensional array 'number of syntactic relations'. The displayed syntactic structure is given in Figure 11. The code of control subroutine is given in Figure 12.

The software is created for testing of the method. The program was prepared for the sentences with some restrictions. The list of restrictions is given in Figure 13. The restrictions may be divided in three types:

1. restrictions, which are conditioned by insufficient computerising of Lithuanian language;
2. restrictions connected with technical data of computer;
3. restrictions, which allow to narrow the area of operating of the program, i.e., the variety of sentences, in order to simplify the practical realization of the software.

With the help of the first restriction one aims to avoid the mistakes, which are foreknown, for example the mistakes which are made due to absence of automatic semantic analysis for Lithuanian language.

For instance, which syntactic function performs gerund in sentences Mačiau skrendant paukštį danguje – I saw the bird flying in the sky and Mačiau lyjant paukštį danguje – I saw the bird in the sky, when it was raining, can only be decided with the help of semantic information. These sentences coincide formally: the ending of both words skrendant (flying) and lyjant (raining) is the same, thus the morphologic forms of both words coincide. The meaning of word only decides the syntactic function of object for the skrendant (flying) and modifier for the lyjant (when it was raining). Therefore in order to avoid the mistakes, which are known in advance, some parts of speech (gerund, half-participle) are rejected (7th restriction in Figure 13).

Restrictions of variety of sentences (1-6 restrictions in Figure 13) allow simplify the software of automatic syntactic analysis, because the morphologic structure of Lithuanian language is very complex and the task to create the software, which could parse any Lithuanian sentence, is very difficult. In order to develop the operational sample more quickly some for parts of sentence are rejected. However, these restrictions don't obtain regarding the method. The proposed method obtains regarding any simple Lithuanian sentence. In order to parse whatever Lithuanian simple sentence one must write additional rules of context free grammar and adequate program code.

```
Private Static Sub Command1_click()
Dim A As Integer            'the position of the word in the sentence
Dim B As Integer            'the current number of morphologic variant of the sentence
Dim C As Integer            'the current number of syntactic variant of the sentence
Dim D As Integer            'the number of words in the sentence
sakinys = Trim(Text1.Text)        'symbol string from text field is assigned to the variable "sentence"
SEGMENTAVIMAS sakinys, ŽodžiųSkaičius, Segmentai()
MORFOLOGIJA Segmentai(), ŽodžiųSkaičius, MorfologijosVariantai(), MorfologijosVariantųSkaičius()
VARIANTAI MorfologijosVariantai(), ŽodžiųSkaičius, MorfologijosVariantųSkaičius(),
                            MorfologiniųSakiniųKiekis, MorfologiniaiSakiniai()
ReDim Preserve MorfologinisSakinys(7, ŽodžiøųSkaičius)
For B = 1 To MorfologiniųSakiniųKiekis     'loop, which submits the variants of morphologic sentences
                            one at a time to the syntactic analysis subroutine
  For A = 1 To ŽodžiųSkaičius          'loop, which singles out one morphologic sentence
    MorfologinisSakinys(1, A) = MorfologiniaiSakiniai(1, A, B)
    MorfologinisSakinys(2, A) = MorfologiniaiSakiniai(2, A, B)
    MorfologinisSakinys(3, A) = MorfologiniaiSakiniai(3, A, B)
    MorfologinisSakinys(4, A) = MorfologiniaiSakiniai(4, A, B)
    MorfologinisSakinys(5, A) = MorfologiniaiSakiniai(5, A, B)
    MorfologinisSakinys(6, A) = MorfologiniaiSakiniai(6, A, B)
    MorfologinisSakinys(7, A) = MorfologiniaiSakiniai(7, A, B)
  Next A
  SINTAKSĖ MorfologinisSakinys(), ŽodžiųSkaičius, SintaksėsVariantai(), SintaksėsVariantųSkaičius()
  VARIANTAI SintaksėsVariantai(), ŽodžiųSkaičius, SintaksėsVariantųSkaičius(),
                            SintaksiniųSakiniųKiekis, SintaksiniaiSakiniai()
  ReDim Preserve SintaksinisSakinys(7, ŽodžiųSkaičius)
  For C = 1 To SintaksiniųSakiniųKiekis
    For D = 1 To ŽodžiųSkaičius
      SintaksinisSakinys(1, D) = SintaksiniaiSakiniai(1, D, C)
      SintaksinisSakinys(2, D) = SintaksiniaiSakiniai(2, D, C)
      SintaksinisSakinys(3, D) = SintaksiniaiSakiniai(3, D, C)
      SintaksinisSakinys(4, D) = SintaksiniaiSakiniai(4, D, C)
      SintaksinisSakinys(5, D) = SintaksiniaiSakiniai(5, D, C)
      SintaksinisSakinys(6, D) = SintaksiniaiSakiniai(6, D, C)
      SintaksinisSakinys(7, D) = SintaksiniaiSakiniai(7, D, C)
    Next D
    GIJOS SintaksinisSakinys(), ŽodžiųSkaičius, AnalizėsRezultatas, SintaksiniaiRyšiai()
    If AnalizėsRezultatas = True Then
    GRAFIKA Segmentai(), ŽodžiųSkaičius, SintaksinisSakinys(), SintaksiniaiRyšiai(), SintaksiniųRyšiųKiekis()
    GoTo sakinioanalizėspabaiga
    End If
  Next C
Next B
sakinioanalizėspabaiga:
End Sub
```

Figure 12. Code of control subroutine for syntactic analysis of Lithuanian sentences

The restriction of third type is related to the technical data of computer (8th restriction in Figure 13). Maximum allowable length of a sentence is indicated in order to avoid the processing of whole text field, which could contain 32000 symbols of Unicode or 64000 symbols of ASCII or ANSI code.

The software is tested with the sentences, which supply the restrictions given in Figure 13.

1. Software will process only simple verbal personal sentences, i.e. the processing of complex and compound sentences, ellipsis and passive is not provided
2. Processing of inversion is not provided too, i.e. processing of sentences for example *Namas mano buvo didelis – House my was big,* when the sentences with such word order are grammatically correct in Lithuanian language.
3. Parts of sentence should be only simple and attribute should be only attributive.
4. Government of adjectives and participles is discounted.
5. Prepositions and abbreviations are not processed.
6. Numerals, words of international orthography, archaisms, appositions, addresses, parenthesis, should not appear in a sentence.
7. The participles are processed as attributes only, because "their meanings as modifiers have no formal features and depend on lexical composition" [7]. Gerund and half-participles will not be processed too, because there is no possibility to decide their syntactic function unambiguously.
8. A maximum limit of 100 words as well as 1000 symbols in a sentence is enforced.

Figure 13. List of restrictions applied to the software

I sample	II sample
Republic periodic	Special periodic
Local periodic	Fiction
Popular periodic	Non-fiction
	Memoir

Figure 14. List of parts of corpus for every sample

4. Estimation of accuracy

The software for automatic syntactic analysis of Lithuanian language is tested with the help of six samples. The test sentences could be divided into three types:

a) The sentences for the first test are taken from all possible sorts of texts, thus the first sample consists of 100 sentences, 50 of which are from Lithuanian corpus (they are selected in concordances) and 50 sentences are taken from grammar of Lithuanian language [7], because there are the sentences especially selected by linguists, which illustrate the parts of sentence.

b) The second test includes four samples (each sample consist of 100 sentences). The first two samples are selected from two different parts of Lithuanian corpus. Their structure is given in Figure 14. Since the search by syntactic features is not provided in corpus, the sentences, which meet the above- mentioned restrictions, imposed on the software, are selected manually. So it is true to say, that not the most appropriate sentences are taken. The concordances are searched in corpus according the given word form and 200 examples are presented, than we checked, which sentence among given 200 sentences satisfies the restrictions. Consequently, the sentences taken from the corpus form the groups with the same word in each sentence and the test looks monotonous.

   The third and fourth samples of this test (of size 100 sentences either) consist of sentences from printed publications. They are divided into two different parts too: books and periodic.

c) The sentences for the third test are selected according the patterns. Lithuanian linguists made 21 patterns for verbal personal sentences [2]. Four patterns with prepositions are not included in the sample, due to the above-mentioned restrictions imposed on the software. We selected 10 sentences for each of other 17 patterns and we get test, which consist if 170 sentences. The sentences are selected from examples given in grammar books and from valence dictionary of Lithuanian verbs. This test reflects the whole structure of Lithuanian language very well, because the patterns in every language are different. The pattern consists of grammatically necessary word forms for a sentence. These word forms are determined by the valence of a verb (predicate). The number of patterns in every language is limited, but according to them one can make an unlimited amount of particular sentences, both the sentences including only words given in model and the sentences supplemented by other words, which are not necessary for the structure of the sentence.

Hereby the test is collected, which consists of 670 sentences: one sample (100 sentences) embraces all types of texts; four different samples by 100 sentences (400 sentences) are composed by selecting the sentences from two different parts of corpus (according to the thematic) and by selecting the sentences from two different parts of printed publications divided into books and periodic; and one sample (170 sentences) is collected according to the patterns.

When the software for automatic syntactic analysis of Lithuanian sentences was tested, the following results were obtained: 629 sentences (out of 670) are parsed correctly. The correctness of the syntactic structure of the sentences approved Lithuanian linguist, dr. of arts E. Valiulytė. The syntactic structures for 41 sentences appeared to be wrong. The mistakes of analysis may be divided into three types:

1. mistakes, which emerged due to absence of semantic information, for example in the sentence *Čia ankstų sekmadienio rytą užsidegė vienas Gėlių gatvės namas – Here in early Sunday morning one house in Flower-street inflamed* the local adverb *čia – here* is treated as modifier of adjective early, and not as adverbial modifier of the verb *užsidegė – inflamed*. Such mistakes happened in 8 sentences.

   The mistakes of this type could be avoided when creating automatic semantic analysis for Lithuanian language i.e. when the information will be possessed that local adverb cannot modify an adjective which has a feature of time and so on.

2. mistakes determined by unsatisfactory supplying of initial data, i.e. when the morphologic analysis gives in the first place very seldom used word form. For example, in the sentence *Ramus jo balsas motiną labai nustebino – His quiet voice surprised the mother* the word Ramas (noun, personal name) is given as the first alternative of adjective *ramus - quiet*. By replacing this word with another word, for example *lout* this mistake disappears. The above-mentioned statements prove, that such mistakes are caused not by program of syntactic analysis. Analogous The mistakes disappears analogous in the sentences *Jis rašo labai paprastai* by replacing the word *labai* (very) the first alternative of which is given: dative case of a noun, with the word *itin* (especially). The syntactic structure is not given for the sentence *Jis rašo eilėraštį Valei* because the noun *Valė* is not entered in the list of words processed by morphologic analysis. Only lemma for this word is given infinitive of verb and word form is indicated following: 2nd person, singular the past tense of verb. Due to the data of the morphologic analysis the mistakes appeared in the structures of 24 sentences.

3. mistakes caused by program of syntactic analysis were in 9 sentences. They appear when in the sentence one of homographs (words identically written but different in morphologic form) is used, for example, coincident endings in genitive case singular and nominative case plural for feminine gender of nouns and adjectives. As instance could be given the sentence *Ateities istorikų laukia nelengvos mūsų Lietuvos studijos – Uneasy study of our Lithuania awaits the historians of the future*. The relation of the word *nelengvos - uneasy* is given with the word *Lithuania* and not with the word *study*. By processing of speech i.e. if the sentence is given not written form, but in speech form (through microphone or telephone) this mistake would disappear because depending on accent one can unambiguously decide the morphologic form of this word. The other way to avoid the mistakes of such type is the creating of automatic syntactic analysis again. Possessing information that the word *easy* can modify the word *study*, but not the word *Lithuania*, one can avoid the mistakes of such type. Another way to solve this problem is design of frequency dictionary of word combinations, which would contain the information that *uneasy study* is more likely than *uneasy Lithuania*.

It is to point out that the semantic features are useful for syntactic analysis. For example, when in a sentence two words are in accusative case and they coincide in their form, i.e. morphologic analysis gives selfsame morphologic data, only semantic features enable to decide, which one of them is object and other is adverbial modifier. As instance of such case could be given the sentence *Vasarą jis padovanojo jai įdomią knygą – He presented her with a book in the summer*. The word *summer* has a feature of time consequently it performs the syntactic function of adverbial modifier. But semantic features help to parse a sentence only in the case when morphologic data are given in the best way (for the word *vasarą – in the summer* only one alternative of word form is given).

However, even the semantic features couldn't help to parse a sentence when the morphologic analysis gives the data not in the best manner. For example, as the first alternative for the word *dieną – in the day* nominative singular masculine of noun is given – *diene* (term of chemistry) and as consequence the feature of time for word form *dieną* is not found in the file of semantic features. Thus in the sentence *Jį man padovanojo Tėvas vienuoliktąją mano gimimo dieną – The father makes me a present of it in my eleventh birth day* both words in accusative case (*jį – it* and *dieną – in day*) are regarded as objects. One must join the morphologic analysis with program of term recognizing [9] to avoid the mistakes of this type.

The results of testing are given in Table 2.

Table 2. Results of testing

Type of test	Characteristic of the sample	Size of sample in sentences	Correct parsed sentenc es	mistakes due to:				precision %
				morphology	semantic	syntax	In total	
I	All sorts of text	100	92	5	2	1	8	92
II	1. periodic in corpus	100	92	3	0	5	8	92
	2. books in corpus	100	90	8	1	1	10	90
	3. printed periodic	100	89	5	4	2	11	89
	4. printed books	100	96	3	1	0	4	96
III	Pattern of sentences	170	170	0	0	0	0	100
In total:		670	629	24	8	9	41	93.88

Generalized it is true to say that software operates with precision 93.88%. The mistakes caused by syntactic analysis program were in 1.34% of sentences. Other mistakes emerged either due insufficiency of semantic information (1.2% of sentences) or due to inaccuracy of morphologic data (3.58% of sentences). These mistakes could be avoided by creating of automatic semantic analysis for Lithuanian language. Namely the automation of semantic analysis should be the main task in computerizing of Lithuanian language.

5. Expansion of applying the method

All the restrictions given in Figure 13 are applied to very program (software) but not to the method. Some restrictions could be removed by using the same method. The most rejected sentences occur due the impossibility to process the prepositions, therefore by example of preposition we will show how the system of syntactic analysis could be expanded and how the processing of prepositions could be included by using the same method.

Prepositional constructions should be described as thread. For example, prepositional construction, which consists of preposition *po (under)* and a nominal in instrumental case, would be following in the BNF (Backus and Naur form) description:

```
<THREAD#PREPOSITION UNDER+INSTRUMANTAL> ::=
     <THREAD#PREPOSITION-UNDER+NOUN-INSTR-SING-MASC> |
     <THREAD#PREPOSITION-UNDER+NOUN-INSTR-SING-FEM> |
     <THREAD#PREPOSITION-UNDER+NOUN-INSTR-PLUR-MASC> |
     <THREAD#PREPOSITION-UNDER+NOUN-INSTR-PLUR-FEM> |
     <THREAD#PREPOSITION-UNDER+PRON-INSTR-SING-MASC> |
     <THREAD#PREPOSITION-UNDER+PRON-INSTR-SING-FEM> |
     <THREAD#PREPOSITION-UNDER+PRON-INSTR-PLUR-MASC> |
     <THREAD#PREPOSITION-UNDER+PRON-INSTR-PLUR-FEM>;
```

Preposition is always located before the noun thus the rule describing the inverted word order is not written in this case.

Pronoun couldn't be extended by any word, therefore the description of its prepositional construction would be the following:

```
<THRAD#PREPORITION-UNDER+PRON-INSTR-SNG-MASC> ::=
          <PREPOSITION-UNDER> <PRON-INSTR-SNG-MASC>;
```

For every alternative of a noun the BNF rule should be written pointing out which words could intervene between the preposition and noun.

```
<THREAD#PREPOSTION-UNDER+NOUN-INSTR-SNG-MASC> ::=
      <PRIEL-PO>
   <INSERTION#PREPOSTION-UNDER+NOUN-I9NSTR-SNG-MASC>
      <NOUN-INSTR-SNG-MASC>;
```

If Insertion consists of several words, which perform the same syntactic function, in the description of Insertion are used angle brackets, for example, agreeing attributes expressed by an adjective or by a participle.

```
<INSERTION#PREPOSTION-UNDER+NOUN-I9NSTR-SNG-MASC> ::=
      [{<AGREEING-ATRIB-ADJ-INSTR-SNG-MASC>}]
      [{<AGREEING-ATRIB-PATICIPLE-INSTR-SNG-MASC>}]
      {<AGREEING-ATRIB-NUMER-INSTR-SNG-MASC>}
            <NONAGREING ATTR-NOUN> |
            <NONAGREING ATTR-PRON>|
<THREAD#NONAGR-ATTR-NOUN+NONAGR-ATTRIB-NOUN>|
<THREAD#NONAGR-ATTR-NOUN+AGR-ATTRIB-ADJEKT>|
<THREAD#NONAGR-ATTR-NOUN+AGR-ATTRIB-PARTICIP>|
<THREAD#NONAGR-ATTR-NOUN+AGR-ATTRIB-NUMERAL>|
<THREAD#AGR-ATTR-ADJ-INSTR-SNG-MASC+ taken ADVERB>|
<THREAD#AGR-ATTR-PARTICIP-INSTR-SNG-MASC+ADVERB>|
```

The subroutine THREAD should be supplemented with processing of new syntactic relations. In block scheme of software (Figure 10) it corresponds to the block of syntactic subroutine.

6. Conclusions

1 The method and algorithm for syntactic analysis of simple Lithuanian sentences are created.

2. The software for syntactic analysis of simple Lithuanian sentences with restrictions is created.

3. The practical use of system of syntactic analysis for Lithuanian simple sentences with restrictions is shown by testing 670 sentences from six sorts of texts and getting the precision 93.88%.

4. A way is described which allows expand the possibilities of applying of method by eliminating the restrictions imposed on the software.

5. It is shown that the system is open and the possibilities are provided for expanding of the system by entering new words and new semantic features.

References

[1] **J. Allen.** Natural Language Understanding. *Amsterdam: The Benjamin/Cummings Publishing Company*, 1987.

[2] **V. Ambrazas (red.).** Dabartinės lietuvių kalbos gramatika. *Vilnius: Mokslo ir enciklopedijų leidybos institutas*, 1997.

[3] **L. Grumadienė, V. Žilinskienė.** Dažninis dabartinės rašomosios lietuvių kalbos žodynas. *Vilnius*, 1997.

[4] **V. Labutis.** Lietuvių kalbos sintaksė. *Vilnius: Vilniaus universiteto leidykla*, 2002.

[5] **E. Rimkutė.** Morfologinio daugiareikšmiškumo ribojimas kompiuteriniame tekstyne. *Daktaro disertacija, VDU Lietuvių kalbos institutas, Kaunas*, 2006.

[6] **D. Šveikauskienė**. Formal description of the syntax of the Lithuanian language. *Information Technologies and Control*, *Vol.*34, *No.*3, 2005.

[7] **K. Ulvydas (red.).** Lietuvių kalbos gramatika. *Tomas* III – *Sintaksė. Vilnius: Mokslas*, 1976.

[8] **T. Winograd.** Language as a Cognitive Process. *Vol.*I: *Syntax. London: Addison- Wesley Publishing Company*, 1983.

[9] **I. Zeller.** Automatinis terminų atpažinimas ir apdorojimas. *Daktaro disertacija, VDU Lietuvių kalbos institutas, Kaunas*, 2005.

[10] **V. Zinkevičius.** Lemuoklis – morfologinei analizei. *Darbai ir dienos* 24, *Kaunas: Vytauto Didžiojo universitetas*, 2000, 245-273.

[11] **В.Н. Билан.** Семантико-синтаксический модуль воспроизводящей инжнерно-лингвитической модели 'Переводчик'. *Поблемы внутренней динамики речевых норм. Минск: МГПИИЯ*, 1982, 174-185.

[12] **Т.А. Грязнухина.** Синтаксический анализ научного текста на ЭВМ. *Киев: Наукова думка*, 1999.

[13] **Е.И. Матвеева.** Автоматический синтаксический анализ с применением сетевой грамматики. *Известия АН Кирг. ССР: Общественные науки. Фрунзе: изд. АН Кирг. ССР*, *№* 2, 1987, 83-88.

[14] **П. И. Сердюков.** Оптимизация алгоритма поиска синтаксических связей. *Международный семинар по машинному переводу. Москва: ВЦП*, 1979, 123-125.

Received May 2007.