

FORMAL MODEL OF NOUN PHRASES IN SERBO-CROATIAN

Goran Nenadić, Duško Vitas
University of Belgrade

Dans cet article, nous présentons une approche de construction de modèles pour la description des noms composés en Serbo-Croate. L'approche est fondée sur le système des dictionnaires électroniques DELA. Nous avons étudié les aspects de construction du DELAC/DEALCF sur les modèles retrouvés ainsi que la reconnaissance des mots composés dans une collection des textes mathématiques.

1. Introduction

In this paper we present an approach to description and modeling of compounds (restricted to noun phrases) in Serbo-Croatian. Since Serbo-Croatian (SC) has rich morphological system, the model is based on initial tagging performed using DELA system of electronic dictionaries (cf. SILBERZTEIN(1993)) for Serbo-Croatian (cf. VITAS(1993)). We have especially studied noun phrases in textbooks in mathematics and computer science, although the proposed model can be used for an ordinary NP in SC. The basic idea is to find out and to define “patterns” for NPs, and then to define formal model that can be used in NLP. The model can be extended so that it covers most NP classes.

The paper is organized as follows: first we briefly discuss the structure of noun phrases in SC and their inflective properties. Possible approaches (especially “patterns”) for NP modeling are presented in section 3. In the next section we define the notion of regular morphosyntactic expressions (RMEs) and the relation for calculating whether an initially tagged sequence of words satisfies a RME. We also present possibilities for NP recognition using RMEs, and relationships between RMEs and DELAC/DEALCF dictionaries for Serbo-Croatian (DELAC-sj/DEALCF-sj dictionaries).

2. Noun phrases in SC

We have initially studied structures of noun phrases (NPs) that occur in textbooks in mathematics, and here we will briefly discuss possible types and describe structures of noun compounds that are most frequent. We will mention some exceptions as well.

We can split our discussion on NPs in two parts: first, we can discuss those NPs that contain at least one fix-form constituent, and second, the rest of NPs - those NPs in which each constituent may appear in different form in the same NP (depending on usage of whole NP). Frozen parts in the first “class” are usually determined by their role in a NP (e.g. partitive “relationship”, like in *solja kafe* ‘cup of coffee’), while in the second, we usually have NPs that consist of determiner(s) and a noun (e.g. *realan broj* ‘real number’).

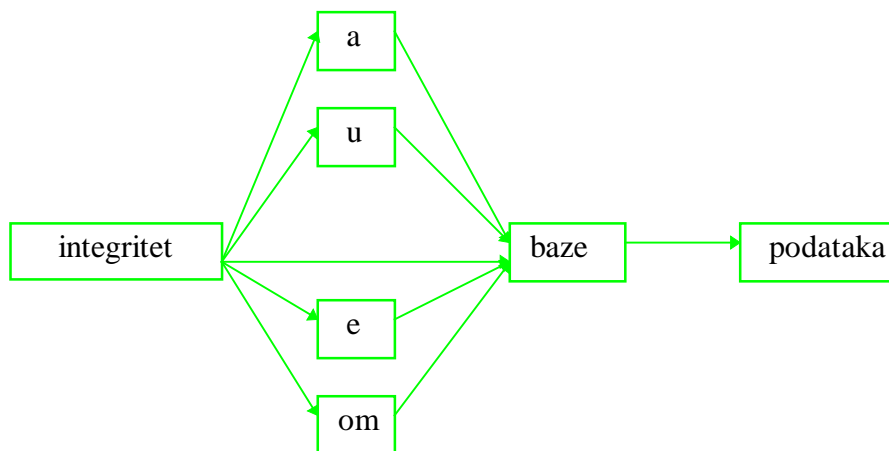
For example, consider the complete inflective paradigm of NPs integritet baze podataka ‘integrity of database’ and realan broj ‘real number’ in singular:

	<i>integritet</i>	<u><i>baze podataka</i></u>		<i>realan</i>	<i>broj</i>
	<i>integriteta</i>	<u><i>baze podataka</i></u>		<i>realnog</i>	<i>broja</i>
	<i>integritetu</i>	<u><i>baze podataka</i></u>		<i>realnom</i>	<i>broju</i>
(1a)	<i>integritet</i>	<u><i>baze podataka</i></u>	(1b)	<i>realan</i>	<i>broj</i>
	<i>integritete</i>	<u><i>baze podataka</i></u>		<i>realni</i>	<i>broju</i>
	<i>integritetu</i>	<u><i>baze podataka</i></u>		<i>realnom</i>	<i>broju</i>
	<i>integritetom</i>	<u><i>baze podataka</i></u>		<i>realnim</i>	<i>brojem</i>

In the first NP (integritet baze podataka), the part that consists of baze podataka appears only in this form (i.e. it is fixed), while only the noun integritet can be found in other forms. We can, then, represent the inflective paradigm (1a) of that NP by a regular expression (RE)

(2) $integritet(\emptyset / a / u / \emptyset / e / u / om) baze podataka$

or by equivalent FSA:



We have a similar notion for noun groups containing prepositions: e.g. the inflective paradigm of the noun group datoteka na disku ‘file on hard disk’ can be represented by:

(3) $datote(ka / ke / ci / ku / ko / ci / kom) na disku.$

On the other hand, one can try to represent the inflective paradigm (1b) in a similar way, but since there is a unique **correspondence** (i.e. agreement) between forms of adjective realan and forms of noun broj, one will get

(4) $real(an broj / nog broja / nom broju / an broj / ni broju / nom broju / nim brojem).$

From this example, it follows that it is not convenient to represent this kind of phenomena (like (1b)) using classical regular expression. In addition, if an NP has more constituents, it is even harder to represent such phenomenon: as an example, consider the inflective paradigm of NP *realan pozitivan broj* ‘real positive number’ in singular:

	<i>realan</i>	<i>pozitivan</i>	<i>broj</i>
	<i>realnog</i>	<i>pozitivnog</i>	<i>broja</i>
	<i>realnom</i>	<i>pozitivnom</i>	<i>broju</i>
(5a)	<i>realan</i>	<i>pozitivan</i>	<i>broj</i>
	<i>realni</i>	<i>pozitivni</i>	<i>broju</i>
	<i>realnom</i>	<i>pozitivnom</i>	<i>broju</i>
	<i>realnim</i>	<i>pozitivnim</i>	<i>brojem</i>

what we can represent by the RE

	<i>real(an pozitivan broj</i>
	<i>/ nog pozitivnog broja</i>
	<i>/ nom pozitivnom broju</i>
(5b)	<i>/ an pozitivan broj</i>
	<i>/ ni pozitivni broju</i>
	<i>/ nom pozitivnom broju</i>
	<i>/ nim pozitivnim brojem).</i>

As we can see, this “class” of NPs, which has no fixed constituents, consists of nouns that are determined by adjective(s) and/or number(s). Using usual notations for denoting nouns (N), adjectives (ADJ), possessive adjectives (PRO_ADJ) and numbers (NUM) we can initially describe this class by the next “**pattern**”

(6) $(ADJ | PRO_ADJ | NUM)^+ N$

which means that, before the noun, there can be arbitrary number of determiners. But, using this pattern, we cannot describe correspondence between forms of constituents: e.g. string **realne pozitivnog brojem* does not represent an NP, although it has the form (6). The scheme (6) is just generic description, and if we want to use it for noun group modeling, it has to be specialized by some constraints and correspondence between constituents of NP.

Besides, there are some noun compounds that have irregular structures, and therefore can be marked as exceptions: e.g.

<i>relacij(a e i u o i om) izmedju</i>	(‘relation between’)
<i>model(∅ a u ∅ e u om) entita i odnosa</i>	(‘entity-relationship model’).

These compounds have frozen parts that are not common to be consider as a “class”(e.g. *izmedju* is a preposition and it makes a compound with *relacija* only in this case and it appears only in mathematical texts).

3. How to represent structures of NPs?

Let us see if it is possible to represent (in a consistent way) structures of NPs that have fixed parts. Consider, for example, a subclass of NPs that contains NPs like *integritet baze podataka* ‘integrity of database’, *sadržaj atributa relacije* ‘content of an attribute of a relation’, etc. In this subclass all NPs have a structure in which the “fixed” part consists of two nouns, both of them appearing in genitive case: *baze* is genitive form (*sing.*) of the noun *baza*, *podataka* is genitive form (*pl.*) of the noun *podatak*, and similarly, *atributa* and *relacije* are genitive forms (*sing.*) of nouns *atribut* and *relacija*, respectively. This “phenomenon” can be described using the following “pattern”:

$$(7) \quad N \quad N_{gen} \quad N_{gen}$$

where the first N denotes first noun (*integritet* or *sadržaj*), which could have complete paradigm, and the last two N_{gen} s represent fixed parts (*baze* and *podataka*, or *atributa* and *relacije*, respectively), which are restricted only to genitive forms.

Similarly, if we consider class of NPs that contain expressions like (3), we can recognize two patterns

$$(8a) \quad N \quad na_{PREP} \quad N_{loc}$$

as for *datoteka na disku* ‘file on hard disk’, and

$$(8b) \quad N \quad na_{PREP} \quad N_{ac}$$

as for *upis na disk* ‘writing on hard disk’, which can be merged into one:

$$(8) \quad N \quad na_{PREP} \quad N_{ac/loc}$$

Using such patterns we can describe subclasses of NPs that have fixed parts, and then classify NPs into them on the basis of their structure. As for an example, we had considered NPs that have the fixed part in genitive (so called ‘genitive constructions’). The following table represents some of the possible classes and ordinary examples of NPs found in mathematical texts:

<i>pattern</i>	<i>example</i>
$N \quad N_{gen}$	nejednakost trougla
$ADJ \quad N \quad N_{gen}$	manipulativni aspekt modela
$N \quad ADJ_{gen} \quad N_{gen}$	operacija prirodnog spajanja
$N \quad N_{gen} \quad N_{gen}$	integritet baze podataka

table 1.

From this, we can conclude that there exist a number of subclasses that can be described abstractly using patterns. The problem is to formalize subclasses in a way that can be used in NLP: we have to define a formal model that can be used not only for description of NPs, but also for recognizing them in an initially tagged text.

4. Regular morphosyntactic expressions

In this section we will define regular morphosyntactic expressions (RME) as a specialization of the notion of regular expressions (cf. HOLUB(1990), AHO(1972)) as defined in formal language theory. First, we introduce a definition of morphosyntactic description (only for nouns and adjectives; this definition can be naturally extended for other grammatical classes — verbs, numbers, etc.)

Morphosyntactic description (MD) is a string with a structure described by the following regular expression:

$$(9) \quad (\{lemma\})? (N | A) [m f o]? [s p]? [n g d a v i l]? [+ -]?$$

As we can see, each element in MD (except (N | A) which denotes grammatical class: N – for nouns, A – for adjectives) is optional. First element (*lemma*) denotes lemma from the dictionary (i.e. dictionary entry), and others denote grammatical categories of gender (m, f, o), number (s, p), case (n, g, d, a, v, i, l) and animation-property (+, –). MD is **general** if it does not contain element *lemma*. MD is **complete** if it contains all optional elements. For example, MDs are:

<i>MD</i>	<i>“meaning”</i>
Ns	a noun in singular
Nsg	a noun in singular, genitive case
baza.Nf	all forms of noun <i>baza</i> (f)
baza.Nfsg-	noun <i>baza</i> in genitive form of singular, (<i>baze</i>)
Ng	a noun in genitive case

Further, we can define **extended MD (EMD)** as a finite **set** of either general MDs (**general EMD**) or complete MDs that all have the same *lemma* (**complete EMD**). EMD is equivalent to the alternation of the MDs it consists of. For example,

<i>EMD</i>	<i>equivalent set of MDs</i>
baza.Nfsg-;Nfnpn-;Nfpi-; Nf;Nfn+;	{ baza.Nfsg- , baza. Nfnpn- , baza.Nfpi- } { Nf , Nfn+ }

Specially, as a result of initial tagging performed using DELAF for Serbo-Croatian (DELAF-sj, cf. VITAS(1993)), one can get a sequence of complete EMDs in an e-text. As an example, consider the initial tagging of the sequence *integritet baze podataka*: 'integrity of database'^{*}

$$(10) \quad \textit{integritet.Nmsn-;Nmsa-; baza.Nfsg-;Nfnpn-;Nfpa-;Nfpv-; podatak.Nmpg-;}$$

* In a tag we omit the textual word: ex. we will write *podatak.Nmpg-*; instead of *podataka.podatak.Nmpg-*; .

It consists of three complete EMDs and represents lexical capability of this sequence of words: it has $2 \times 4 \times 1 = 8$ possible lexical interpretations.

Let us, now, define a **match**-relation on the set of all MDs:

match(D_1, D_2) is true **iff** each element of MD that is present in D_2 is present in D_1 too, i.e. one can get D_2 by omitting some elements (i.e. constraints) in D_1 .

In that case, we can also say that D_1 is **more restrictive** than D_2 . Consider examples:

match (Nsg , Ns) = T	match (baza.Nfg- , baza.Nf) = T
match (baza.Nf , N) = T	match (N , baza.Nf) = F
match (Nsg , Nf) = F	match (knjiga.Nf , baza.Nf) = F

It is not difficult to show that **match**-relation is reflexive partial order on the set of MDs.

Further, let us extend **match**-relation so that the first argument can be an EMD:

match(E_1, D_2) is true **iff** there exists a MD D_1 in E_1 such that **match**(D_1, D_2):

$$(11a) \quad \mathbf{match}(E_1, D_2) = \bigvee_{D_1 \in E_1} \mathbf{match}(D_1, D_2)$$

For example,

match(*baza.Nfsg-;Nfnpn-;Nfpa-;Nfpv-; , Ng*) = **T**,

since

match(*baza.Nfsg-; , Ng*) = **T**.

Finally, let us define **regular morphosyntactic expression (RME)** recursively:

RME is either a textual word (lexemme) or an EMD, or
if R_1 and R_2 are RMEs then RMEs are $R_1R_2, R_1 | R_2, (R_1), R_1?, R_1^*, R_1^+$.

For example, RMEs are

$N ? Ng Ng$ $A? baza.Nfsg-; podatak.Nmpg-;$ $A^+ N$

Specially, we will consider **simple RMEs** - RMEs that are concatenations of textual words and/or MDs.

Now, let us extend **match**-relation to cover an initially tagged sequence of words ($t_1 \dots t_n$) as the first argument of the relation, and a simple RME ($r_1 \dots r_n$), as the second:

a sequence $t_1 \dots t_n$ of initially tagged words **matches** a simple RME $r_1 \dots r_n$ **iff** each t_i matches corresponding r_i , i.e.

$$(11b) \quad \mathbf{match}(t_1 \dots t_n, r_1 \dots r_n) = \bigwedge_{i=1}^n \mathbf{match}(t_i, r_i)$$

For example, if we consider textual sequence *integritet baze podataka* (and the corresponding initial tagging (10)) and the simple RME $N Ng Ng$ we can calculate:

$$\begin{aligned}
 & \mathbf{match}(integritet.Nmsn-;Nmsa-; baza.Nfsg-;Nfpn-;Nfpa-;Nfpv-; podatak.Nmpg-, N Ng Ng) \\
 &= \mathbf{match}(integritet.Nmsn-;Nmsa-; , N) \wedge \\
 (12) \quad & \mathbf{match}(baza.Nfsg-;Nfpn-;Nfpa-;Nfpv-; , Ng) \wedge \\
 & \mathbf{match}(podatak.Nmpg-, Ng) \\
 &= \mathbf{T} \wedge \mathbf{T} \wedge \mathbf{T} = \mathbf{T}
 \end{aligned}$$

From this, we can conclude that textual sequence *integritet baze podataka* matches RME $N Ng Ng$, or that it satisfies the pattern $N N_{gen} N_{gen}$.

5. Compounds and RMEs

5.1 Representing NPs using RMEs

Using RMEs we can now represent formal models of NPs that have fixed parts. For NPs given in table 1, we can define the following RMEs that uniquely represent subclasses of NPs:

<i>pattern</i>	<i>RME</i>
$N N_{gen}$	$N Ng$
$ADJ N N_{gen}$	$A N Ng$
$N ADJ_{gen} N_{gen}$	$N Ag Ng$
$N N_{gen} N_{gen}$	$N Ng Ng$

table 2.

Similarly, the RME $N \underline{na}$ ($Na | Nl$) denotes NP class that corresponds to the (8). These RMEs represent not only model of an NP class, but, additionally, they describe the complete inflective paradigm of that class (all parts that are not present in a MD can be variable), i.e. a RME describes all possible textual occurrences of an NP that has the appropriate structure.

From this, we can conclude that using RMEs we can describe NP classes that contain fixed parts. Now, the question is whether we can use these definitions for compounds recognition in an initially tagged text.

5.2 Compound recognition using RMEs

Suppose that we have an initially tagged text (as a result of performing DELAF look-up). As we have seen, it contains a sequence of (complete) EMDs (see (10)). We will show that it is possible to recognize those sequences of EMDs that satisfy (i.e. **match**) some RME.

First, we show how to transform the **match**-relation on sequences of **complete** EMDs and MDs (that represent a pattern) into the classical regular-expression matching. To do that, we transform each MD from the pattern into an ‘*equivalent*’ RE (a **canonical** MD) by adding (as regular expressions) all parts that are not present in the MD. For example, for the morphosyntactic description **Nsg** (*a noun in singular, genitive case*) the canonical MD is $\{\textit{lemma}\}.\text{N}[\textit{mfo}]\text{sg}[+-]$, in which we have added RE for denoting lemma ($\{\textit{lemma}\}$), and character classes for gender ($[\textit{mfo}]$) and animation-property ($[+-]$), which are not present in the MD. In this context, we can say that a MD and its canonical MD describe the same set of complete EMDs. Now, since we have to determine the value of the **match**-relation on a **complete** EMD (i.e. on an initial tag), it is enough (and equivalent) to check (using regular expressions) if this EMD matches given canonical MD (that represents part of NP). For example, if we want to determine whether **match**(baza.Nfsg- ; Nsg) is true, we have to check whether the *string* **baza.Nfsg-** satisfies the regular expression defined by the canonical MD $\{\textit{lemma}\}.\text{N}[\textit{mfo}]\text{sg}[+-]$:

$$\text{match}(\text{ baza.Nfsg- } , \text{ Nsg}) = (\text{ baza.Nfsg- }) \sim (\{\textit{lemma}\}.\text{N}[\textit{mfo}]\text{sg } [+-])$$

where \sim is classical RE-matching operator. Naturally, if we have an EMD that represents more than one MD, we convert the calculating of **match**-relation into the appropriate disjunction (as in (11a)): e.g.

$$\begin{aligned} \text{match}(\text{ baza.Nfsg-;Nfnp-; } , \text{ Nsg}) &= (\text{ baza.Nfsg- }) \sim (\{\textit{lemma}\}.\text{N}[\textit{mfo}]\text{sg } [+-]) \\ &\quad \vee (\text{ baza.Nfnp- }) \sim (\{\textit{lemma}\}.\text{N}[\textit{mfo}]\text{sg } [+-]) \\ &= \text{ T } \vee \text{ F } = \text{ T}. \end{aligned}$$

Generally, we calculate:

$$(13) \quad \text{match}(\text{ EMD } , \text{ MD}) = \bigvee_{\text{D} \in \text{EMD}} (\text{ D } \sim \text{ canonical}(\text{MD}))$$

Further, when a pattern is a sequence of MDs, we have to calculate the appropriate conjunction (as in (11b)).

From this, we can conclude that if we want to use our NP definitions for recognizing textual sequences that make NPs, first we have to “translate” models from RMEs to equivalent REs. It is obvious that we can automatically convert an RME into an equivalent RE: e.g., for table 2 we can get

<i>RME</i>	equivalent RE ({lem} is abbr. of {lemma})
N Ng	$\{\textit{lem}\}.\text{N}[\textit{mfo}][\textit{sp}][\textit{ngdavid}][+-] \{\textit{lem}\}.\text{N}[\textit{mfo}][\textit{sp}]\text{g}[+-]$
A N Ng	$\{\textit{lem}\}.\text{A}[\textit{mfo}][\textit{sp}][\textit{ngdavid}][+-] \{\textit{lem}\}.\text{N}[\textit{mfo}][\textit{sp}][\textit{ngdavid}][+-] \{\textit{lem}\}.\text{N}[\textit{mfo}][\textit{sp}]\text{g}[+-]$
N Ag Ng	$\{\textit{lem}\}.\text{N}[\textit{mfo}][\textit{sp}][\textit{ngdavid}][+-] \{\textit{lem}\}.\text{A}[\textit{mfo}][\textit{sp}]\text{g}[+-] \{\textit{lem}\}.\text{N}[\textit{mfo}][\textit{sp}]\text{g}[+-]$
N Ng Ng	$\{\textit{lem}\}.\text{N}[\textit{mfo}][\textit{sp}][\textit{ngdavid}][+-] \{\textit{lem}\}.\text{N}[\textit{mfo}][\textit{sp}]\text{g}[+-] \{\textit{lem}\}.\text{N}[\textit{mfo}][\textit{sp}]\text{g}[+-]$

table 3.

Similarly, the RME $N \underline{na} (Na | NI)$ can be converted into the equivalent RE:

(14) $\{lemma\}.N[mfo][sp][ngdavid][+ -] \mathbf{na} \{lemma\}.N[mfo][sp][\mathbf{al}][+ -]$.

This RME actually represent a **local grammar** (cf. SILBERZTEIN(1994), NENADIC(1997)) that can be used for lexical disambiguation of sequence of words. It means that we can omit some MDs from an EMD if the sequence satisfies the local grammar. For example, the initial tagging of the sequence *datoteka na disku* ‘file on hard disk’

(15) $datoteka.Nfsn-;Nfpg-; \mathbf{na} disk.Nmsd-;Nmsl-;$

which represents four lexical possibilities, matches the RME (14) since

$\mathbf{match}(datoteka.Nfsn-;Nfpg-; , \{lemma\}.N[mfo][sp][ngdavid][+ -]) = \mathbf{T}$ and
 $\mathbf{match}(disk.Nmsd-;Nmsl-; , \{lemma\}.N[mfo][sp][\mathbf{al}][+ -]) = \mathbf{T}$,

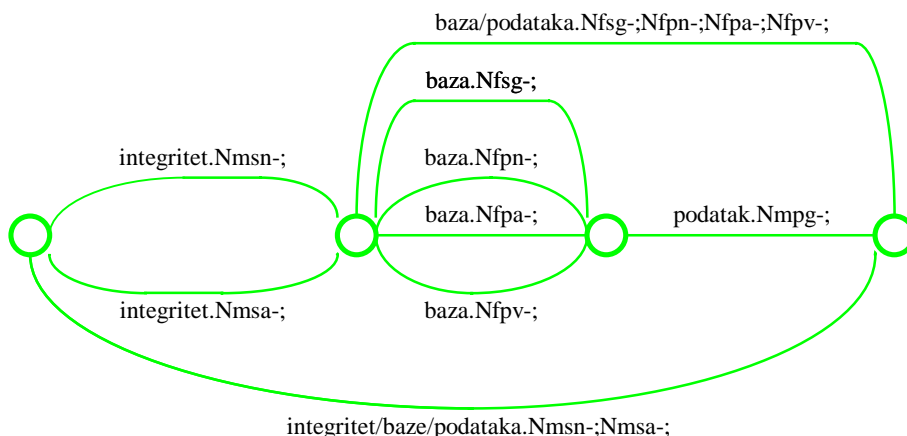
and, consequently, we can **exclude** MD *disk.Nmsd-;* from *disk.Nmsd-;Nmsl-;* and this way disambiguate (15):

(15') $datoteka.Nfsn-;Nfpg-; \mathbf{na} disk.Nmsl-;$

Using classic RE-tools (ex. **lex**), it is now possible to implement procedures for recognition of strings of words that satisfy a formal RME model. To do this, we have to implement calculation of **match**-relation. Calculating **match**-relation (as defined by (11b) and (13)) and procedures for recognition of “genitive constructions” are implemented (NENADIC(1997)): when a string of initially tagged words matches one of RMEs, it is marked as **potential compound**, and corresponding EMD and canonical form (i.e. lemma or dictionary entry) of compound are assigned to it. Additionally, all procedures that recognize patterns (i.e. classes of NPs), can work in parallel on the same text. If we consider example (10) and the value of **match**-relation (12) again, we can **mark** this sequence of three words as one (noun) compound (of the class $N N_{gen} N_{gen}$), with an EMD that is identical to the EMD of first constituent (since, the first N is *center* of NPs that satisfy pattern $N N_{gen} N_{gen}$, see below). This way, from (10) we can get the EMD

$integritet/baze/podataka.Nmsn-;Nmsa-;$

which denotes compound as noun group in singular, that can be either in nominative or accusative case. This new possible “reading” can be added to the FSA that represents the sequence:



The way of choosing the EMD (i.e. gender, case etc.) of a compound is determined by the “center” of the NP, which is defined for each pattern. For example, for patterns represented in table 2 we can define:

<i>pattern</i>	<i>center</i>
$N N_{gen}$	first constituent (N)
$ADJ N N_{gen}$	second constituent (N)
$N ADJ_{gen} N_{gen}$	first constituent (N)
$N N_{gen} N_{gen}$	first constituent (N)

table 4.

The system of NP classification can be used for entry modeling for **DELAC-sj** electronic dictionary of compounds (cf. NENADIC(1997)), as well as for automated generation of **DELACF-sj** from DELAC-sj and DELAF-sj (cf. NENADIC(1997)). For example, if we denote NP class $N N_{gen} N_{gen}$ as class 01.03, then the DELAC-sj entry that corresponds to compound *integritet/baze/podataka* could be:

integritet/baze/podataka.N01.03

From this definition, one can calculate complete paradigm of the compound, and generate the appropriate part of DELACF-sj:

integritet/baze/podataka, integritet/baze/podataka.N01.03:Nmsn-;Nmsa-;
integriteta/baze/podataka, integritet/baze/podataka.N01.03:Nmsg-;Nmpg-;
integritete/baze/podataka, integritet/baze/podataka.N01.03:Nmsv-;Nmpa-;
integriteti/baze/podataka, integritet/baze/podataka.N01.03:Nmpn-;Nmpv-;
integritetima/baze/podataka, integritet/baze/podataka.N01.03:Nmpd-;Nmpl-;Nmpi-;
integritetom/baze/podataka, integritet/baze/podataka.N01.03:Nmsi-;
integritetu/baze/podataka, integritet/baze/podataka.N01.03: Nmsd-;Nmsl-;

From this, we can see that NP patterns can be used for description of DELAC-sj entries primarily in the phase of defining models for NP representation, and even for generation of NP inflective paradigm. Additionally, the system can be extended as automated help for

completion of DELAC-sj with new compounds that satisfy a pattern (since, we can even handle compounds that are entered for the first time in a text), or for searching for new NP patterns in texts.

6. Conclusion

In this paper, we have presented an approach to formal modeling of noun phrases in SC. The model allows us to find out whether a sequence of words satisfies a pattern that represents a NP. Using this approach, which is an extension of the notion of local grammars, one can define “classes” of NPs and realize procedure for NP recognition in an initially tagged text by calculating specific relations on a sequence of words. This way, we can disambiguate a sequence, and retrieve complex compounds as well (e.g. for terminological databases). Also, the approach can help in automated completing of e-dictionary of compound words, since it can handle NPs that have been entered for the first time in a text.

REFERENCES:

- AHO A., ULLMAN J. (1972), “*The Theory of Parsing, Translation and Compiling*”, vol. no 1., Prentice-Hall, New Jersey.
- HOLUB A. (1990), “*Compiler Design in C*”, Prentice-Hall, New Jersey.
- NENADIC G. (1997), “*Algorithms for compound word recognition in mathematical texts and applications*”, MSc thesis, Faculty of Mathematics, University of Belgrade (in Serbo-Croatian)
- SILBERZTEIN M. (1993), “*Dictionnaires électroniques et analyse automatique de textes: le système INTEX*”, Masson, Paris.
- SILBERZTEIN M. (1994), “*INTEX: a Corpus Processing System*”, Proc. of COLING 94, ACL, Tokyo.
- VITAS D. (1993), “*Mathematical Model of Serbo-Croatian Morphology (Nominal Inflection)*”, PhD thesis, Faculty of Mathematics, University of Belgrade, (in Serbo-Croatian)