Studia Linguistica 46(1) 1992

Vladimir Pericliev

A REFERENT GRAMMAR TREATMENT OF SOME PROBLEMS IN THE BULGARIAN NOMINAL PHRASE

Abstract. The paper addresses some problems of the Bulgarian nominal phrase within the framework of Referent Grammar, a grammatical model used for computer analysis, generation and translation in the SWETRA project (Swedish Computer Translation Center at the Department of Linguistics and Phonetics, Lund University). In particular, the computer implementation of agreement, the choice of articled forms, and the movement of clitic words in the Bulgarian nominal phrase are discussed.

1. Introduction

Referent Grammar (henceforth, RG) is a grammatical model developed by Bengt Sigurd (1987, 1988) which can be expressed in the Definite Clause Grammar (DCG) formalism (Pereira & Warren 1980, Clocksin & Mellish 1981), and used for automatic analysis, generation and translation of sentences and texts. So far, this model has been successfully used for the description of (and translation between) fragments of languages like Swedish, English, French, Georgian, and Samoan within the framework of the SWETRA project (Swedish Computer Translation Center at the Department of Linguistics and Phonetics, Lund University).

In this paper, I will try to show how some problems of the Bulgarian nominal phrase can be handled within this framework. In particular, I will be concerned with constructions referred to in RG as the Noun Phrase Head (nph), i.e. phrases like the English *good books*, *all my five black cats*, etc., containing prepositioned non-homogeneous modifiers to a noun.

I shall begin the discussion with a general, and informal, description of the Bulgarian nph (Section 2). In the sections to follow, this general description will be made more precise, and it will be shown how the rules proposed can be implemented in the RG formalism (DCG). In particular, in Section 3 I will treat problems of gender and number agreement between adjectival modifiers and the noun head. Section 4 focuses on the choice of an articled form in the nph. Finally, Section 5 shows how the movement of clitic attributes within the nph can be handled.

2. The Bulgarian noun phrase head

The noun in the Bulgarian nph can be modified by a number of adverbial and adjectival words. In the first class belong the negation particle *not* (abbreviated as neg), restrictive adverbs (adv_restr, e.g. *only*), adverbs (adv) for quantity, whereas in the second class belong quantifiers (quant), or totality markers (e.g. *all*), demonstrative pronouns (pron_dem), possessive pronouns (pron_poss), cardinal numerals (num_c), ordinal numerals (num_o), and adjectives (adj).

Following in (1) are typical Bulgarian examples of nph's consisting of pairs of words, a modifier and a noun:

(1)	(a)	neg + noun:	*ne knigata* 'not the book'
	(b)	adv_restr + noun:	*samo knigi* 'only books'
	(c)	quant + noun:	*vsički knigi* 'all books'
	(d)	pron_dem + noun:	*tezi knigi* 'these books'
	(e)	pron_poss + noun:	*negovata kniga* 'his book'
	(f)	num_o + noun:	*parvata kniga* 'the first book'
	(g)	num_c + noun:	*pet knigi* 'five books'
	(h)	adv + noun:	*mnogo knigi* 'many books'
	(i)	adj + noun:	*debela kniga* 'thick book'

As in most languages, the Bulgarian noun may have more than just one modifier. The relative word order of these modifiers, when they occur in a sequence, is quite strict in the language (stylistic nuances aside). In fact, it corresponds to the above numeration in the sense that the modifier in the first construction (i.e. neg) occurs first, the modifier in the second (i.e. adv_restr) occurs second, etc. Or, in general, the relative order of a nph sequence is a + b + c + d + e + f + g + h + noun.

Each of these modifiers is, moreover, optional. Thus, in a familiar rewriting notation, where optional categories stand between square brackets, we can formulate the following rule, describing the relative word order and optionality of modifiers in the Bulgarian nph:

(2) nph → [neg] [adv_restr] [quant] [pron_dem] [pron_poss]
[num_o] [num_c] [adv] [adj] noun

This rule thus describes nph's such as *knigi* 'books', *tezi mnogo knigi* (literally) 'these many books', *ne samo vsički tezi naši parvi pet hubavi knigi* (literally) 'not only all these our first five nice books', etc.

The rule in (2) however, is deficient in at least two respects inasmuch as its eventual computer implementation is concerned.

In the first place, it lacks some information (e.g. it does not say that the adjectival categories agree with the noun; nor does it specify how to choose articled forms in the nph). Furthermore, due to its intended generality of formulation, it contains certain inaccuracies (e.g. it does not account for certain co-occurrence restrictions in the nph). The following sections refine this rule by supplying necessary information or imposing appropriate constraints, whenever Rule 2 tends to overgenerate.

3. Agreement in the noun phrase head

Before proceeding with agreement considerations, a few words are in order about the representation of linguistic knowledge within the RG framework. In what follows, I will assume some background in Prolog and, specifically, the DCG formalism.

RG states its rules in DCG format, using the usual generative rewrite-style notation. As a result of the processing, structures are built specific to its (translational) needs. Specific to RG is the building of both a constituent representation (o-representation) and a "universal" functional representation (f-representation), the f-representation serving as input to the translation process. Another essential feature of RG is the building of a referent identification structure. Also, taking advantage of Prolog's possibilities, the rules are generally formulated to be “reversible”, i.e. applicable for both analysis and generation. More information on the architecture of RG can be found e.g. in Sigurd (1987, 1988). Suffice it to mention that our rules will also be reversible, and provide further explanation about the structure of the lexicon and rules in RG.

RG represents dictionary entries as Prolog facts. Usually each lexical item is a ten-place predicate with a name which is a derivative from *lex* (e.g. *slex* for Swedish lexeme, *blex* for Bulgarian lexeme, etc.). (Naturally, the number of predicate places is not inherent to RG and may vary both with respect to different languages and different word classes.) The first three places in the predicate *lex* are reserved for the graphic form of the lexeme, its meaning (expressed in “machinese English”), and its word class, respectively. The other slots in the *lex* predicates vary with the word class, but normally contain information about the word class’s morphological features (e.g. Gender, Number, Definiteness as far as nouns are concerned) or semantic type. Some sample Bulgarian dictionary entries are presented below in (3):

(3) a) blex(moliv,m(pencil,sg),n,ma,sg,indef,_,_,_,_).
b) blex(moliva,m(pencil,sg),n,ma,sg,def(noful),_,_,_,_).
c) blex(molivat,m(pencil,sg),n,ma,sg,def(ful),_,_,_,_).
d) blex(molivi,m(pencil,pl),n,_,pl,indef,_,_,_,_).
e) blex(moliva,m(pencil,pl),n,_,pl_c,indef,_,_,_,_).
f) blex(loš,m(bad,_),a,ma,sg,indef,_,_,_,_).
g) blex(mnogo,m(many,_),adv,_,_,_,_,_,_,quantity).
h) blex(tozi,m(this,sg),pron_dem,ma,sg,def,_,_,_,_).

Some of the abbreviations are self-explanatory: n = noun, a = adjective, adv = adverb, ma = masculine, sg = singular, indef = indefinite. Others express information specific to Bulgarian: def(noful) and def(ful) (cf. (3)(b) and (c)) stand for definite nouns which have “non-full” or “full” forms respectively (the masculine singular nouns in Bulgarian have special, “full”, forms for the case when they are subjects in the sentence, and other, “non-full”, forms for the other functional roles). The abbreviation pl_c in (2)(e) stands for plural “cardinal” (the Bulgarian masculine noun has a special plural form, pl_c, when it is modified by a cardinal numeral, and another, “normal” form, abbreviated as pl, elsewhere).

By way of illustration, we may look at a very simple rule RG might use for the description of a nph:

```
(4)     nph(r(R,M3,D,N,G), h(M3,M1,M2)) →
              pron_dem(M1,G,N,D), adj(M2,G,N,D1),
                                        n(M3,G,N,D2).
```

Roughly, the rule in (4) states, as in generative grammar, that a nph consists of a pron_dem, adj and n. The structure to be built as a result of the processing is stated within the brackets to the left of the arrow, and consists, in this particular case, of the referent structure, r(R,M3,D,N,G), and the structure of the head, h(M3,M1,M2). The terms in capital letters are variables (as in Prolog) whose values will be computed in the unification process. The variable R, stands for a referent number, ranging over the set of natural numbers; the numbered variables M1, M2, M3 stand for the (machinese English) meanings of the words in the string parsed, and G, N, D for the features Gender, Number and Definiteness, respectively.

In order to have this rule executed by Prolog, we only need to add some "dictionary rules", linking the above rule with the lexicon described above, e.g.:

```
(5)     pron_dem(M,G,N,D) →
                      [Prd],{blex(Prd,M,pron-dem,
                                          G,N,D,_,_,_,_)}.
        adj(M,G,N,D) → [Adj],{blex(Adj,M,a,G,N,D,_,_,_,_)}.
        adv(M,G,N,D) → [Noun],{blex(Noun,M,n,
                                          G,N,D,_,_,_,_)}.
```

In order to parse the nph *tozi loš moliv* 'this bad pencil', the following question is then required:

```
        ? nph(Referent,Head,[tozi,loš,moliv],[ ]).
```

Prolog will then respond:

```
Referent = r(_00B0,m(pencil,sg),def,sg,ma)
Head = h(m(pencil,sg),m(this,sg),m(bad,_0490))
```

building the referent and head representation (the numbers with underscores are variables as they are internally represented by Prolog; the referent variable R remains uninstantiated in this case since, obviously, it makes sense only in contexts larger than the constructions we are dealing with here).

We now turn to agreement. In Bulgarian, as in many other languages, the adjectival modifiers agree in number and gender with the noun head. This agreement property of the nph can be easily stated in terms of RG (DCG) by writing the same variables for Number and Gender both in the adjectival words and in the noun in one rule (in Prolog different occurrences in one clause of variables with the same name are instantiated to the same value). In fact the rule in (4) above ensures this agreement by declaring the same variables, N and G, standing for Number and Gender respectively, in the pron_dem, adj and n goals.

After these remarks, we can restate the generative rule in (2) as a RG rule, which also accounts for agreement (for the sake of brevity, we omit the usual rules for optionality of classes):

```
(6)     nph(r(R,M10,D,N,G),
                  h(M10,M1,M2,M3,M4,M5,M6,M7,M8,M9)) →
        /* 1 */ neg(M1),
        /* 2 */ adv_restr(M2),
        /* 3 */ quant(M3,G,N,D1),
        /* 4 */ pron_dem(M4,G,N,D2),
        /* 5 */ pron_poss(M5,G,N,D3),
        /* 6 */ num_o(M6,G,N,D4),
        /* 7 */ num_c(M7,G,N1,D5),
/* 8 */  { ifthenelse(M7\ = [ ],N2 = pl_c,N2 = N) },
/* 9 */  adv(M8,Qt),
/* 10 */ adj(M9,G,N,D6),
/* 11 */ n(M10,G,N2,D7).
```

Agreement is expressed, as shown above, with variable propagation. Gender agreement is ensured by declaring the same variable, G, throughout the relevant goals.

Notice, however, a certain detail in the number agreement. As already mentioned, the Bulgarian masculine noun has two plural forms: one for the case when this noun is modified by a cardinal numeral (a plural cardinal form, pl_c), and another "normal" form (pl) in the remaining cases. Thus, we say *pet moliva(pl_c)* 'five pencils', but *hubavi molivi(pl)* 'good pencils', using different noun suffixes in both cases.

The rule in (6) implements this situation in the following way. A predicate with obvious semantics, *ifthenelse* (system predicate in Arity Prolog), is used as a Prolog call in curly brackets (line 8), to check whether a cardinal numeral is realized in the terminal string parsed. Should it be realized (i.e. the goal M7\ = [ ], where M7 is the meaning of the numeral, succeeds), the number variable of the noun, N2, is assigned to pl_c; in case no such numeral occurs (M7\ = [ ] fails), the same variable N2 is shared with N, or in other words, is passed to the "normal" agreement pattern, which may be in the sg or pl (but not pl_c), depending on the value which the variable N has up to this moment been instantiated to. (We note that the less compact statement of the condition in question in the standard Prolog syntax will be: {M7\ = [ ],N2 = pl_c; M7 = [ ],N2 = N }.)

Bulgarian in general offers no further problems concerning gender and number agreement. Indeed, there exist commonly occurring nouns with "dual" gender, with which adjectival words may optionally agree either in the masculine or feminine (such as the Bulgarian word for *dust*); in addition, there exist nouns with "dual" number, where the agreement is optionally in the singular or plural (such as the Bulgarian equivalents for *mouth* or *door*). However, such nouns are extremely limited in number (about a dozen), and hence may be represented as different entries in the dictionary, thus eliminating the need for changing the above agreement technique in order to incorporate these additional facts into the grammar.

As far as words for professions are concerned (such as *professor*, *minister*, etc.), which are troublesome in some languages, these cause no agreement problems in Bulgarian within the type of constructions we are discussing. This is so since they have constant gender within the Bulgarian nph and may be modified only by adjectival words inflected correspondingly. E.g. we may say *loš(Gender:ma) profesor(Gender:ma)* ‘a bad professor’, but not *“*loša(Gender:fe) professor(Gender:ma)* ‘a bad (woman) professor’. (Outside the nph, however, the situation would be different: these words, acting as copula objects, may have subjects in all genders, and may be referred to by pronouns having different genders in other constructions, etc.)

4. The choice of articled forms

The Bulgarian definite article originates from the weak (postpositional) demonstrative pronoun in Old Bulgarian. Being originally an enclitic, this pronoun was positioned in the nominal phrase according to the law of the Indo-European enclitic, i.e. second in the noun phrase, immediately after the first word. In Modern Bulgarian, this enclitic is no longer a separate word but rather has turned into a morpheme, linked to the preceding word. Hence, if a nph is definite, the very first adjectival word in the nominal sequence is inflected for definiteness; the remaining adjectival words are inflected for indefiniteness. In the indefinite nph, on the other hand, all words in the nominal phrase are inflected for indefiniteness.

One way to implement these facts in the grammar is to check, after each execution of a goal involving an adjectival word, to see whether this word is definite or not, and pass the corresponding information on to the next goals. These checks can be stated in RG in terms of conditions in curly brackets after each relevant rule. However, this solution does not seem to be the best one in that it will unnecessarily complicate the grammar (in the above case e.g. six conditions of this type will have to be added). Thus it is preferable to write just one predicate to do the job. Having just one predicate is furthermore advantageous owing to the fact that the same predicate can be used in other parts of the grammar, involving this type of agreement.

More concretely, we add the following goal 12 to the rule in (6):

```
/* 12 */ { choose_article([D1,D2,D3,D4,D5,D6,D7], D) }
```

In its first argument, goal 12 collects in a list structure the values of the variables for D(efiniteness) in the order they are computed in Rule 6, viz. D1, D2, D3, ... These values may be def, def(ful), def(noful), indef or the empty list [ ], where def stands for definiteness of feminine nouns, def(ful) and def(noful) for the two forms of definiteness of masculine nouns, and indef for indefiniteness of nouns of all genders. Some of the variables D1, D2, D3, ... could be instantiated to the empty string because some adjectival classes may not appear in the string to be parsed, in which case they match, in our grammar, the empty list. The goal's second variable argument D, whose value is to be obtained as a result of further computations, should say whether the nph is definite or indefinite, and the computed result should be returned to the referent structure.

Now we may define a predicate with the name *choose_article*, having two arguments:

```
choose_article(List,X)
```

The predicate *choose_article* takes as input the list of the already computed values for D1, D2, D3, ... Then, from this list, each occurrence of the empty string is deleted, with a view to ensuring the possibility of determining the Definiteness value of the first occurring adjectival word in the parsed string. (In the definition listed below, the deletion is done by the predicate *del*.) The newly obtained list as a result of the deletion is passed to another predicate, *choose_art*. The first clause of *choose_art* is the limiting condition, warranting the termination of its recursive computation. Its second clause isolates the head of the list, returning its value as the Definiteness value of the nph, and saying that the tail of this list should contain only elements with the value indef. The full definition of these predicates is as follows:

```
(7)      choose_article(List,X):-
                 del([ ],List,List1),
                 choose_art(List1,X).
         choose_art([ ],_).
         choose_art([H|T],H):- !,
                 choose_art(T,indef).
         del(_,[ ],[ ]).
         del(A,[A|T],L):-!,del(A,T,L).
         del(A,[B|T1],[B|T2]):- del(A,T1,T2).
```

Thus, in the way described, we have ensured that our nph grammar accepts (generates) only well-formed nominal sequences insofar as the choice of article is concerned, e.g. *vsičkite(def) naši(indef) hubavi(indef) knigi(indef)* 'lit.: the-all our good books', or *vsički(indef) naši(indef) hubavi(indef) knigi(indef)* 'lit.: all our good books', but not, for instance, **vsički(indef) našite(def) hubavi(indef) knigi(indef)* 'lit.: all the-our good books', or **vsičkite(def) našite(def) hubavi(indef) knigi(indef)* '(lit.: the-all the-our good books'.

5. Handling clitic attributes

So far, we have not mentioned the fact that the noun in the Bulgarian nph may be modified, in addition to the already discussed attributes, by some attributes not in concord. These are the so-called "short possessive pronouns" (pron_poss_sh) and "short reflexive pronoun" (pron_refl_sh). Let us now examine this unexplored possibility.

Examples of such uses are: *vsičkite ni(pron_poss_sh) knigi* 'all our books', and *vsičkite si(pron_refl_sh) knigi* 'all books of my/your.../ their own'. (Additionally, of course, we may use the full form (pron_poss) of the possessive pronoun to express the same thing.)

The treatment of clitics is not straightforward, the basic reason being their "movement" within the nph. Indeed, as enclitics, these forms, if they occur, always come immediately after the first adjectival word in the nph (and, in this sense, the clitics' position is invariable). However, since the rule in (6) contains optional adjectival categories, which may or may not be realized in the parsed terminal string, the clitics will "move", or have different potential positions within the rule.

These potential positions, of which a clitic may occupy just one, will be as shown below in Diagram 1:

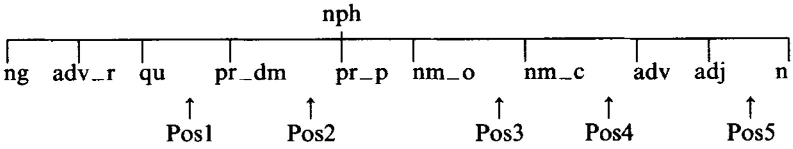

Diagram 1

Besides the basic problem of movement (in the above sense), the incorporation of clitics into the grammar will have to account for the following further facts: (1) if a clitic occurs, the immediately preceding word is necessarily definite; and (2) the short possessive pronoun and the full possessive pronoun are mutually exclusive, i.e. only one of them can occur in a grammatical nph. The latter constraint explains why, in the diagram above, no potential position of a clitic after a full possessive pronoun (pr_p) is envisaged.

The behaviour of clitics described may be implemented in the following way. First, we add the following rules to our nph grammar:

```
(8)     clitic(Pr_poss_sh) → pron_poss_sh(Pr_poss_sh).
        clitic(Pr_refl_sh) → pron_refl_sh(Pr_refl_sh).
        clitic([ ]) → [ ].
```

The first two rules state that a clitic is either a short possessive pronoun or a short reflexive pronoun; the third rule says that the clitic forms may be absent from the parsed string. (For the sake of simplicity, the variables propagating the meaning of clitics are omitted.)

As a next step, we insert into the rule in (6) – at each of the potential positons shown above – two goals, having the form:

```
        clitic(Clitic,M), { check_clitic(Clitic,D) }
```

which, roughly, state that a clitic may occur in the parsed string only if there exists an immediately preceding word such that its variable for the feature Definiteness is D, where D stands for "definite".

The corresponding definition of the predicate *check_clitic*, triggering this behaviour, is as follows:

```
(9)     check_clitic([ ],_):- !.
        check_clitic(_,def).
        check_clitic(_,def(ful)).
        check_clitic(_,def(noful)).
```

The first clause of the predicate *check_clitic* succeeds if no clitic appears in the parsed string. The next three clauses succeed if a clitic does appear, and one of the three alternatives is valid: the preceding word's variable for Definiteness is assigned to the values def, def(ful) or def(noful). The predicate fails in case there is no word preceding a clitic, or this word is indefinite.

The above amendments describe some of the behaviour of clitics: they warrant the fulfilment of the word order constraint on clitics, and the constraint on the preceding word (constraint 1 above). As far as co-occurrence constraint 2 is concerned, it is only partially fulfilled: so far, in expanding the category clitic into two disjunctive classes, pron_poss_sh and pron_refl_sh, we have only captured the fact that the latter classes do not occur in the same position in the terminal string.

Thus, as a third step in our implementation, we shall have to impose a further restriction forbidding the co-occurrence of a full possessive pronoun and a clitic. We can easily do that by inserting a condition on the execution of the goal searching for a full possessive pronoun, stating that such a pronoun may be realized in the terminal string if no preceding clitic has been found in this string (see the ifthen goals below).

With these amendments, a fragment of the rule in (6) including just some relevant categories, will have the following form:

```
(6')    (Fragment of (6))
        nph(r(R,M4,D,N,G), h(M4,M1,M2,M3)) →
              quant(M1,G,N,D1),
              clitic(Clitic1), { check_clitic(Clitic1,D1) },
              pron_dem(M2,G,N,D2),
              clitic(Clitic2), { check_clitic(Clitic2,D2) },
              pron_poss(M3,G,N,D3),
‘{ifthen(Clitic1\=[ ],M3 = [ ]),ifthen(Clitic2
                                        \=[ ],M3=[ ])},
n(M4,G,N,D4),
{ choose_article([D1,D2,D3,D4], D) }.
```

Now the rule has exactly the behaviour we need. At first glance, it might be suspected, judging by the placement of the clitic subgoals at each potential clitic position, that the rule overgenerates, admitting unacceptable strings with repetitive occurrences of a clitic, such as *_vsičkite ni tezi ni knigi_ ‘lit.: the-all _our_ these _our_ books’, or *_visčkite ni ni knigi_ ‘lit.: the-all _our our_ books’. However, a clitic subgoal may succeed with a terminal realization of a clitic not more than once. This is so due to the fact that the occurrence of a clitic requires the preceding occurrence of a definite word, and such a word may occur not more than once, the latter being guaranteed by the choose_article predicate. Put differently, the correct behaviour of the rule is achieved by the interaction of the subgoals clitic and check_clitic, on the one hand, and choose_article, on the other.

Our treatment may raise a further question. The potential clitic positions were shown when we have a “flat rule” like Rule 6′, where the nph expands into such nonterminals which do not themselves expand into further nonterminal categories. In this case, these positions could be directly stated in the rule rewriting the nph. But what should be done in non-flat, or rules of several levels, in which the dependents of nph expand into further nonterminals, and, what is important, a clitic is positioned _in-between_ the latter nonterminals? In such cases we cannot state positions directly into the nph rule. This is, however, only a seeming problem: the same technique of clitic treatment is applicable here as well, and we only have to state the goals involving clitics at the appropriate, next lower level. Thus, for instance, if the nph can be rewritten say into a demonstrative and an adjectival _phrase_ where a clitic is admissible between its adjectives, we should state the clitic goals not in the nph, but in the rule expanding the adjectival phrase. The following (very simplified) rules illustrate the situation:

```
nph → pron_dem, clitic, { check_clitic },
        adj_phrase, { choose_article }.
adj_phrase → adj1, clitic, { check_clitic },
                adj2, { choose_article }.
```

A final remark. Our approach to clitic movement, amounting to stating potential positions followed by defining constraints on their eventual occupation, is quite similar to the approach to movement adopted in the Generalized Word Order Grammar (Sigurd 1990). Clearly, what I call "movement" here is distinct from the construal of the term in the latter system, and generally in linguistics; nevertheless, the computational treatment of these phenomena may be similar.

6. Conclusion

In this paper, I have tried to show how phenomena like agreement, choice of articled forms, and clitic movement in the Bulgarian nph can be described within the framework of RG. Our analysis is still, needless to say, both inexhaustive (it does not cover a number of further nph constructions) and imprecise in certain respects (the grammar still overgenerates somewhat). However, no further major grammar refinements will be necessary since the Bulgarian nph presents no further principled difficulties.

References

CLOCKSIN, W. F. & MELLISH, S. 1981. *Programming in Prolog*. Berlin: Springer.

PEREIRA, F. & WARREN, D. H. D. 1980. Definite Clause Grammars for language analysis – a survey of the formalism and a comparison with Augmented Transition Networks. *Artificial Intelligence* 13, 231–278.

SIGURD, B. 1987. Referent Grammar (RG). A Generalized Phrase Structure Grammar with built-in referents. *Studia Linguistica* 41, 115–135.

SIGURD, B. 1988. Using Referent Grammar (RG) in computer analysis, generation and translation of sentences. *Nordic Journal of Linguistics* 11, 129–150.

SIGURD, B. 1990. Implementing the generalized word order Grammars of Chomsky and Diderichsen. *Proceedings of COLING*, vol 3, 437–439. Helsinki.

Vladimir Pericliev
Mathematical Linguistics Dept.
Institute of Mathematics, bl.8
1113 Sofia, Bulgaria