Computer Science Journal of Moldova, vol.**4**, no.2(11), 1996

Some tools to implement linguistics applications oriented to the Romanian language*

E.Boian, S.Cojocaru, L.Malahova

Abstract

In the article the Romanian Spelling Pack is proposed which is a set of dynamic link libraries work under MS Windows. They implement checking a Romanian word against the vocabulary base; search through word base for one close to a given word; Romanian hyphenation; user vocabulary extension; Romanian word inflection. These procedures can be used in processing Romanian texts in Windows environment and in writing interface and tutorial programs for the Romanian language. As an example of the approach, Romanian spelling checker for MS Word 6.0 is described.

1 Introduction

One of the problems with a natural language processing software is how to integrate in various environments and how to develop an application for a specific platform. We propose so-called Romanian Spelling Pack (RomPW) which is represented under Windows by several DLLs (dynamic link libraries).

This article is a survey of the results obtained in the Institute of Mathematics of Academy of Sciences of Moldova in the computational morphology (linguistics) during last five years.

Development of the structure of dictionaries had posed many problems. Two main ones were those of the compact representation of the

*This work was partially supported by the Soros Foundation of Moldova, project No.1 ED OPEN 96

vocabulary including all flexions and of the fast search in the vocabulary. We discuss them in the sec. 2.

The Hyphenation function, the Checking function, the function of Romanian words inflection, the Vocabulary support function are compound parts of the Romanian Spelling Pack. The components of this Pack are described in the sec. 3.

We will show its integration into MS Word 6.0 as the word processing environment (Romanian Spelling Checker) in the sec. 4.

The screen font is elaborated in code MS 1250 (Central and Eastern Europe).

RomPW is a developing system, and the perspectives of its development are presented in the final sec. 5.

2 Effective vocabulary representation

Being among highly inflexional languages, Romanian makes really difficult the problem of compact representation of its vocabulary V. One of the well-known approaches here is to separate roots R and endings E.

Using the definition of binary decomposition specified in [1, 2], one evidently can choose various ways for constructing such decompositions: it is quite possible that $R = V$ and all endings are empty words, or vice versa, when there is a single root — empty word, but all the elements of V serve as endings. If V is the vocabulary of word-forms for a language, there is some hope that taking a natural decomposition into E and R the above method leads to the reasonable map. It means that list L of all the possible values of subsets $f(r)$ would be not so large (comparatively with the size of V). In this case it would be sufficient to keep with the every root r only the index of its subset $f(r)$ in a list L, so the necessary memory for the vocabulary would consist of two main parts: memory for roots set R (plus memory for index for every root) and memory for the list L of possible sets of endings.

The starting point for this approach was book [3], where main part of Romanian inflective words were classified according to the methods of creation the flexions. There were 100 groups for masculine nouns,

273 for verbs etc in the book, and about 30000 words with their group numbers were listed. The classification was made from the linguistical point of view, and, for example, the accents were taken into account. Nevertheless, this classification was useful and have lead to the idea to introduce the special grammar formalizing word-forms production. Using these grammar rules, we can formalize the process of creation of the decomposed vocabulary.

The above method of the decomposition is based on the knowledge about the morphological group of the given word. Nevertheless it is necessary to have the possibility to include a new set of word-forms for the given item without this knowledge. We need to detect the group number dynamically.

First of all the word-forms themselves should be obtained. A special program can be elaborated to facilitate this boring work. The information concerning part of speech (verb, noun etc) and, may be, something else (e.g. gender) can be obtained interactively. Then deep linguistic analysis (and may be some additional questions (usually about alternation or suffixes)) permits to predict the possible structure of the base-form. The second step is to detect roots in word-forms. For each root we store on the hard disk the number of the corresponding ending set. To increase the efficiency, we divide the vocabulary into pages and use a hash table to access a page. Ending sets are represented as bitmaps, the hash table and endings themselves are stored in RAM. Also in RAM a small vocabulary of about 600 most frequently used Romanian words is stored.

As the matter of fact, we have two vocabularies: a constant one and a user's private one. The user's vocabulary can be expanded by the user.

3 Composition of the Romanian Spelling Pack

Romanian Spelling Pack consists of the following components:

- hyphenation function,
- checking function,

- function of Romanian words inflection,
- vocabulary support function.

We discuss these components of RomPW in more detail in further subsections.

3.1 Hyphenation function

The Hyphenation function does not depend on the orthography variant of a word and does not use vocabularies. The function has two parameters. The first parameter is a pointer to a zero-terminated string (C-string) containing a word to be hyphenated. The second parameter is the pointer to the buffer reserved for the result. Syllables in the result are separated by hyphens (minuses).

The algorithm maximally takes into consideration classical rules of word division into syllables, which base on letters' phonetic significance [4]. The specific character of Romanian language does not permit completely formalize them. The problem of diphthongs and threephthongs, which is the most difficult one in the process of word division into syllables for Romanian language, is solved only for some specific situations. Namely, when the diphthongs are at the beginning or at the end of the word and for some cases in the middle of the word.

There is one more difficult problem, which we can solve only partially – hyphenation of compound words. There are compound words of the two types:

1) words, which are always written with hyphen: *baba-oarba*, *prim-ministru*, *anglo-franco-italian*;

2) words, consisting of several words, which are written as a single word: *feldmareşal*, *concertmaistru*, *binefacere*.

We can not correctly divide all the words of the second group yet, for the lack of the necessary information. For example, the words *bi-ne-fa-ce-re*, *bu-nă-vo-in-ţă* are divided by our algorithm correctly, but such compound words, as *feldmareşal*, *concertmaistru*, are divided

incorrectly. Each word of the first group is processed as a group of independent words, between which the hyphen is already known: *ba-ba-oar-ba*, *prim-mi-nis-tru*,*an-glo-fran-co-i-ta-li-an*.

Test showed, that 70% of the words in the texts from the scientific and art literature are divided correctly.

3.2 Checking function

The Checking function is a function which checks a word against the base and activates a dialogue if the word is not found in the base. The dialogue proposes the following possibilities [8]:

- to edit the word manually;
- to ask for similar word search the vocabulary (a suggestion);
- to ask for the word deletion;
- to check the correctness of a word;
- to declare the word as a good one for the rest of the session;
- to add the word to the user private vocabulary (using the function of Romanian word inflection);
- to stop spell checking.

3.3 Function of Romanian words inflection

The function of Romanian words inflection inflects Romanian words by special word-forming procedures. These procedures take into consideration the division of the set of words into parts of speech. It needs additional information to decline nouns and adjectives and to conjugate verbs because they generate a lot of inflections (12 for nouns, 20 for adjectives, 35 for verbs). Pronouns, articles and numerals were entered in the vocabulary data base in special mode.

Six parts of speech of eleven can be inflected by special procedures. Nouns, adjectives, articles, numerals and pronouns are declined in accordance with case, number and form. Verb is conjugated in accordance

with tense, mood, person etc. The rest – adverb, preposition, conjunction, particle and interjection – is invariable and because these parts of speech are not so numerous they are introduced in vocabulary directly without change by inflecting procedures.

We investigate in more details inflecting regularities of nouns, adjectives, verbs, articles, numerals and pronouns.

3.3.1 Nouns and adjectives

We have determined the criteria to classify nouns and adjectives in three inflexion groups: automated, partial automated and irregular. To inflect a word it is necessary to know vowel and consonant alternations and affix series.

The affix series tables, the alternations set and their admissible combinations [5, 6, 7] form the inflexion program base. The vowel and consonant alternations are characteristic for noun and adjective inflexion. Absolute regular alternations are divided, in its turn, into the two groups: automatic absolute regular, when flexions are produced without alternations or with consonant alternations and semiautomatic for which it is characteristic the vowel and consonant alternations.

After the affix separation in the base word the automatic absolute regular alternations rules specified for nouns and adjectives are applied (for example, the consonant alternations for masculine nouns, the vowel alternations $ea \rightarrow e$, $ia \rightarrow ie$ for nouns and adjectives etc.) The partial regular alternations require a detailed context analysis, sometimes there is a need to ask the user for additional information. Words with irregularities form a separate set. They are emphasized apriori and processed in a special way.

The word for inflection is entered then the speech part of this word is determined. If the word is a noun, then its gender and its number are specified. For the word its base form (indefinite form, nominative case and singular number) is specified. Division of the word into the root and the affix is made in the following way. The affixes specified in [5, 6, 7] are arranged in the decreased order of their lengths. If proposed the noun for flexion belongs to the set of irregular words then

all the flexions immediately appear on the screen, else for each affix a special procedure of noun inflexion is selected. It finds in the word the affix in the decreased order of length. If it coincides with one of the affixes, then the corresponding procedure of inflexion is called. The word affix serves as a distinctive criteria. If the key–affix belongs to the absolute regular set of nouns or adjectives (to be inflected without the user's interference) then the specified word is declined in accordance with the inflexion model founded. If the key–affix belongs to the set of partial regular nouns or adjectives it is necessary to select the appropriate alternations rules from several possible variants. In this case it is necessary to initiate a dialogue where the user can select the suitable variant or to add a new one. To simplify the user's work the inflexion programs generate all possible variants of application of the alternation rules. Some of these words may seem strange, but this situation make the selection easy. For example if we inflect the word *dulap*. The procedure suggests variants for the neuter noun, plural number, nominative case: **dulap***uri*, **dulap***e*, (as *fir* – *fire*). The user selects the suitable word **dulap***uri*. After that the corresponding procedure produces the other necessary flexions.

The word inflection program shows all generated forms on the screen, and the user can edit them before writing them to the vocabulary data base.

In [9] we obtained some statistical data which indicates the inflexion process automatization degree. Analyzing these data one can conclude that 88% of nouns and adjectives can be declined automatically and only 12% need a dialog.

3.3.2 Verbs

The verb inflexion procedure forms a words-form list in accordance with the infinitive of a verb [5, 6]. This list contains Imperative, Participle (for which the gender is characteristic) flexions and verb flexions of those tenses only, for which changes in accordance with person is determined (i.e. Indicative Present, Conjunctive Present, Imperfect, Perfect Simple).

The rules of conjugation were formulated for all schemes of verb conjugation [5, 6]. Of course, there are exceptions, which are updated separately, such as auxiliary, defective verbs etc. The verbs set is divided into 56 schemes of verbs conjugation. It is necessary to note that within each conjugation scheme the subcase limits (i.e. within the mentioned 10 cases limits), which reflect the conjugation specific features of some verb subgroups, are selected.

The full information about including a verb into one or an other grammatical group is determined in procedural way, from the verb ending in the infinitive, but for the verbs of grammatical groups I-st and IV-th the information is necessary whether the verb conjugates with or without a suffix. As the matter of fact it is the information about impersonality of the verb. The function depending on the answer generates 40 or 12 flexions respectively. The answers are obtained from the dialog between the user and the system. Knowing this information the inflexion process is finished automatically.

Consequently we concluded that the formalization of cases and subcases is a rather difficult problem, moreover if we consider the development and extension of a natural language, usage of different forms with the same grammatical categories (for example: *comenzi* and *comanzi*), then it is clear that the results of the program's work must be examined and corrected if it is necessary by the user and only after that the flexions are included into a dictionary.

3.3.3 Articles, numerals and pronouns

These parts of speech produce a various irregular forms of flexions. They are grouped into sets of articles, numerals and pronouns correspondingly. Every element of mentioned sets is inflected as irregular noun a special way.

The article set is divided into tree groups: indefinite, demonstrative and possessive one. For example, the indefinite article has the following forms:

Cases	Sing. masc.	Sing. fem.	Pl. masc. fem.
N/A	un	o	nişte
G/D	unui	unei	unor

The set of numbers can be divided into two main groups. The first group contains indeclinable elements such as some of simple cardinals (for ex. *cinci*, *treizeci*, *unsprezece* etc.) ordinals, collectives (for ex. *tuspatru*). The second group contains declinable ones such as some of simple cardinals (for ex.*zece*, *sută*, *milion* etc.), and fractionals (for ex. *cincime*, *jumătate*, *sfert*) which are declined as nouns; multiplicative numbers (for ex. *îndoit*, *însutit*, *înmiit*) which are declined as adjectives. For some of numerals the gender is characteristic. For example, the cardinal number *unu* has adjectival value for the masculine – *un băiat* and for the feminine – *o fată*; ordinal numbers have specific ends *-lea* for the masculine gender and *-ea* or *-a* for the feminine one, correspondingly.

Analogous as numbers the set of pronouns are grouped into the two subsets. The first subset contains indeclinable elements such as possessive and emphatic pronouns (for ex. *meu*, *nostru*, *lor* etc. and *însumi*, *însevă* etc. correspondingly). They are declined in accordance with their gender, person and number. For example, emphatic pronouns have the following forms:

Person	Sing. masc.	Sing. fem.	Pl. masc.	Pl. fem.
I pers.	însumi	însămi	înşine	însene
II pers.	însuţi	însăţi	înşivă	însevă
III pers.	însuşi	însăşi	înşişi	înseşi, însele

The second subset contains the most part of pronouns. Every type of pronouns has a specific mode of declination. Personal pronouns have not only the usual forms for nominative and vocative (for II person, singular and plural numbers) cases, but they have special forms for dative and accusative stressed and unstressed cases (it is proper for reflexive pronouns also) and have not the genitive case. For example, personal

pronouns *eu, noi* have the next forms(I person, singular number):

Cases	Stressed unstressed	Singular	Plural
Nominative		eu	noi
Genitive		–	–
Dative	stressed unstressed	mie îmi, mi	nouă ne
Accuzative	stressed unstressed	mine mă	noi ne
Vocative		–	–

The III person of personal pronoun besides the case forms has the gender ones. For example, *el* for masculine and *ea* for feminine gender.

For every kind of declinable pronouns we obtain different number of flexions: for courteous pronouns – 2 (nominative-accusative-vocative and genitive-dative cases for II person (the negative ones also)) and 5 – for III person; for interrogative-relative pronouns: *cine* has 2 flexions, *cât* has 6 flexions, *care* has 8 flexions. The last two cases have not only the case and number flexions, but they have gender ones also.

The very interesting case is the demonstrative and indefinite pronouns. They can be used as independent part of proposition (themselves) and as attribute (pronominal adjective). Declination of these pronouns generate 8 flexions. The most part of these forms (for pronouns and pronominal adjectives) coincide. For example, the demonstrative pronoun and the pronominal adjective *acelaşi*:

Cases	Sing. masc. neut.	Sing. fem.	Pl. masc.	Pl. fem. neut.
N/A	acelaşi	aceeaşi	aceiaşi	aceleaşi
G/D	aceluiaşi	aceleiaşi	aceloraşi	aceloraşi

One can note that if for adjective and noun declination or for verb conjugation the user has to indicate the base word with its necessary morphological category to obtain all flexions, for article, numeral and

pronoun declination it is sufficient to show one of flexions, on base of which all the flexions appear on the screen with indication of the type of specified word. For example, if user indicates the word *lor*, he points out the pronoun part of speech, the pronoun inflexion procedure shows on the screen all flexions for the personal pronouns *ei, ele* for which *lor* is a flexion for genitive and stressed form dative cases masculine and feminine genders, III person, plural number and possessive pronoun *lor*.

The described inflecting process was used to create a vocabulary of about 65.000 base words using lexicographical sources [10 – 19].

3.4 Vocabulary support function

The Vocabulary support function is executed by a separate interactive program to maintain the vocabulary data base [8]. This program has the following functions:

- to initialize the base (to create a new empty base);
- to check the base integrity;
- to compress the base;
- to add a word;
- to add words from a file;
- to delete a word;
- to search for a word;
- to search for similar words;
- to make triades;
- to show the hash table;
- to show information about pages;
- to show last words from each page;

- to output words page by page;
- to output words in the single list;
- to add a synonym/translation for a word;
- to add synonyms/translations from a file;
- to delete a synonym/translation for a word;
- to show synonyms/translations for a word;
- to stop a execution of program.

These Vocabulary support functions are implemented by DLLs which represent the vocabulary data base support. The DLL does not use vocabularies. Other DLLs are of higher level and can be called directly from applications.

Two functions need explanation – "to compress the base" and "to make triades".

Compressing means that all words stored into the user vocabulary which is not such effective are moved to the main vocabulary.

"Triades" is the table of all three-letter combinations existing in the vocabulary entries. It is organized as an array of bitmaps. We use it when generating suggestions.

"Pages" are not printing pages but those vocabulary data base pages described above in sec. 2.

Synonym and translation support is the experimental part of our project. Now it is not included into the distributed version.

The output list of words can be used later to reconstruct the whole vocabulary from scratch. We had also used it to collect some statistical information.

4 Romanian spelling checker for MS Word 6.0: an example of the application of RomPW

MS Word 6.0 includes Word Basic (a specialized subset of Visual Basic) which is used as the MS Word macro-programming language. It per-

mits to program applications integrated into MS Word. Using Word Basic we implemented a scanner which reads words from a selected part of text in the opened window and then calls the word checking function from the Romanian Spelling Pack.

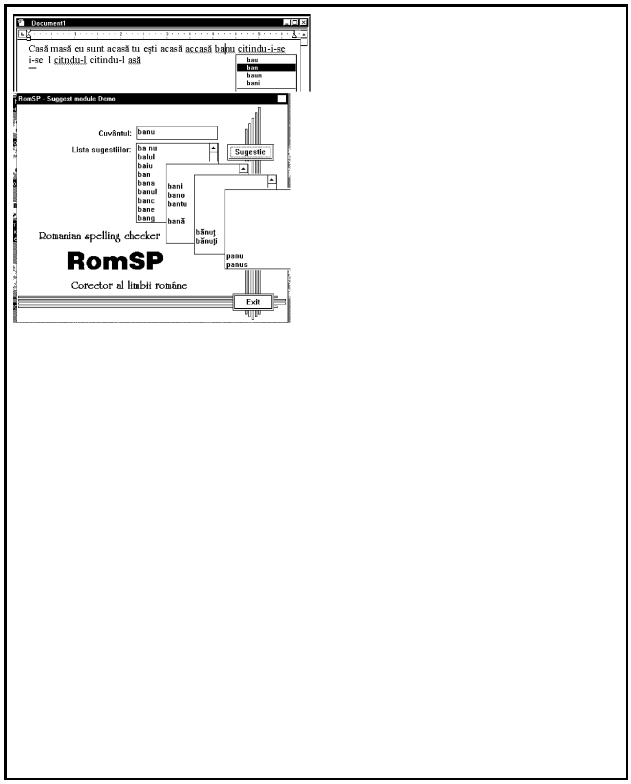
The checking function takes as a parameter a string (the word to check). This word is searched through the vocabulary. The function returns a string starting with the result code and can contain after it the replacing word. If the word was not found into the vocabulary the checking function described above calls the dialogue.

In the case of the vocabulary extension, there is a possibility to inflect the word and to include it with all its flexions.

A special problem in scanning the Romanian text is that of words with hyphens, when a particle is appended to a word. Some of such particles can appear as separate words, some of them can appear only in this situation. Hyphens are also used to make compound words. To check such cases the scanner uses a forward preview and, finding the hyphen, it appends the next word to the current one. This combination is transferred to the checking function which analyzes these situations.

The advantage of the used technique of the spelling checker integration into the word processor is that the specialized library can process more complicated situations of the specific language. We had seen it on the example of hyphen processing in Romanian words. Another example is our suggestion algorithm which uses information specific for the Romanian language and can find 2–3 errors.

We can compare our approach with the standard approach of Microsoft spelling tools implemented for the Romanian language by MorphoLogic (Hungary) and Software ITC (Cluj, România).



Fugure 1. The comparison of our approach with the standard approach of Microsoft spelling tools implemented for the Romanian language by MorphoLogic (Hungary) and Software ITC (Cluj, România).

We see in Figure 1 the construction *citindu-i-se* was erroneously declared wrong, and the separate *i-se* and even the letter *l* were skipped as good.

Our suggestion algorithm [1, 2] gives more suggestions as more adequate for the Romanian language.

We can add to the user vocabulary all the flexions for a word.

5 Conclusions

.

The Romanian Spelling Pack is the resource added to Windows and can be used in any application through the corresponding interface.

- The Romanian Spelling Pack may be used to implement spelling checkers for text processing programs.
- The Word inflection function may be used also for implementing of tutorial programs for the Romanian language.
- The Hyphenation function may be used in text editor processing programs.

References

[1] S.Cojocaru, M.Evstiunin, V.Ufnarovsky. Detecting and correcting spelling errors for the Romanian language. Computer Science Journal of Moldova, Vol.1, No.1(1), 1993

[2] S.Cojocaru, M.Evstiunin, V.Ufnarovski. Romanian spelling-checker. Studies in Informatics and Control, Vol.3, No.1(1), March 1994

[3] A.Lombard, C.Gadei. Dictionnaire morphologique de la langue roumaine [The morphological Romanian language dictionary]. Bucureşti, 1981 (French)

[4] V.Demidova, T.Verlan. An approach to the word division into syllables for Romanian language. Computer Science Journal of Moldova, Vol.4, No.1(10), 1995, pp.59–68.

[5] E.Boian, A.Danilchenco, L.Topal. The automation of speech parts inflexion process. Computer Science Journal of Moldova, Vol.1, No.2(2), 1993

[6] E.Boian, A.Danilchenco, L.Topal. Automation of word-forming process in the Romanian language. Studies in Informatics and Control, Vol.3, No.1(7), March 1994

[7] E.Boian, S.Cojocaru. The inflexion regularities for the Romanian language. Computer Science Journal of Moldova, Vol.4, No.1(10), 1995, pp.40–58.

[8] A. Colesnikov. The Romanian spelling checker ROMSP: the project overview. Computer Science Journal of Moldova, Vol.3, No.1(7), 1995, pp.40–54.

[9] S. Cojocaru. The Romanian lexicon: software, implementation, utilization. The Language and Technologies. Dan Tufiş – editor, Publishing house. The Romanian Academy, Bucharest, 1996, pp.37–39 (Romanian)

Lexicographical sources

[10] Dicţionarul explicativ al limbii române [The Romanian language explanatory dictionary]. Bucureşti, 1975 (Romanian)

[11] Supliment la dicţionarul explicativ al limbii române [Supplement to the Romanian language explanatory dictionary]. Bucureşti, 1988 (Romanian)

[12] F.Marcu. Mic dicţionar de neologisme [The small neologism dictionary]. Bucureşti, 1985 (Romanian)

[13] Dicţionar ortografic cu elemente de ortoepie şi morfologie [The orthographical dictionary with elements of orthoepy and morphology]. Kishinev, 1991 (Romanian)

[14] F.Şuteţ, E.Şoşa. Dicţionar ortografic al limbii române [The orthographical Romanian language dictionary]. Bucureşti, 1994 (Romanian)

[15] ORTO. Dicţionar ortografic cu elemente de ortoepie şi morfologie. Kishinev: Red. Principală a Enciclopediei Sovietice Moldoveneşti, 1990, 608 p.

[16] V. Bucur. Scurt dicţionar de teorie a evidenţei contabile rus-român şi român-rus [The short Russian-Romanian and Romanian-Russian dictionary on bookkeeping]. Kishinev, 1992 (Russian and Romanian)

[17] Mic dicţionar rus-român de termini economici [The small Russian-Romanian economical terms dictionary]. Kishinev, 1991 (Russian and Romanian)

[18] C.Tănase. Dicţionar de terminologie financiară rus-român [The Russian-Romanian financial terms dictionary]. Kishinev, 1992 (Russian and Romanian)

[19] M.Carauş. Mic dicţionar de termini de economie [The small economical terms dictionary]. Kishinev, 1990 (Russian and Romanian)

Elena Boian graduated from the Department of Cybernetics of the Kishinev University in 1979. She obtained her doctoral degree in Computer Science from Department of Cybernetics of the Kiev State University in 1992 (Ukraine). Since 1979 she has been working at the Institute of Mathematics of the Academy of Sciences of the Republic of Moldova. Her present position is a senior researcher. Dr. Elena Boian published more than 30 scientific papers. Her research interests include functional programming, parallel programming, compiler construction, natural language processing.
phone: (373–2) 738073; fax: (373–2) 738027
e–mail: 22lena@math.moldova.su

Svetlana Cojocaru graduated from the Department of Cybernetics of the Kishinev University in 1974. She obtained her doctoral degree in Computer Science from the Institute of Cybernetics in Kiev (Ukraine) in 1982. Since 1974 she has been working at the Institute of Mathematics of the Academy of Sciences of the Republic of Moldova. Her present position is leading researcher. Dr. Svetlana Cojocaru published more than 30 scientific papers. Her research interests include formal grammars, compiler construction, natural language processing, computer algebra.
phone: (373–2) 738073; fax: (373–2) 738027
e–mail: 22svet@math.moldova.su

Liudmila Malahova graduated from the Department of Cybernetics of the Kishinev University in 1970. Since 1970 she has been working at the Institute of Mathematics of the Academy of Sciences of the Republic of Moldova. Her present position is leading engineer. Ludmila Malahova published more than 20 scientific papers. Her research interests include compiler construction, natural language processing.
phone: (373–2) 738058; fax: (373–2) 738027
e–mail: 21mal@math.moldova.su