# A Procedure for Word Derivational Processes Concerning Lexicon Extension in Highly Inflected Languages

**Klára Osolsobě, Karel Pala, Radek Sedláček, Marek Veber**

Faculty of Informatics, Masaryk University Brno
klara@ernest.phil.muni.cz, {pala,rsedlac,mara}@fi.muni.cz

**Abstract**

The aim of this paper is to describe an efficient tool (I_PAR) for a supervised and semi-automatic extension of a lexicon or morphological database and its easy updating. We will present the underlying algorithms and their implementation that are general enough to capture the main word-forming processes (both inflectional and derivational). They are designed for languages with a rich inflectional morphology, such as Slavonic languages, particularly Czech. The implementation is partly based on the ideas presented in the earlier paper by Klímová and Pala (2000)

## 1. Introduction

Our aim has been to develop a user-friendly computer system that, based on user's set(s) of examples, discovers rules in the form of derivational patterns and tries to apply them to the existing static Czech stem dictionary (containing approximately 150,000 items). The main feature of this approach is that the actual stem dictionary or morphological database becomes dynamic, i.e. the word forms are not stored as lemmata, but they are obtained from their roots by applying the derivational or inflectional rules. They allow us to extend and complete the general lexicon in a regular and possibly exhaustive way.

## 2. Procedure for morphological database extension

The main idea is the following: a user, typically a linguist, is able to formulate a set of examples capturing a particular derivational process based on some inflectional and derivational paradigms he or she is interested in. But there can be problems with its generalisation to some extent. One of the most important reasons for these complications is a limited amount of data that can be considered at once. Thus, the first step consists in collecting the examples, the more, the better. The user is allowed to specify negative examples as well. Typically, on the morphological level, an example corresponds to one or more derivational steps and consists of the appropriate basic word and the respective derived one(s), e.g. *učit* (teach) – *učitel* (male teacher) – *učitelka* (female teacher) – *učitelstvo* (teachers as a societal group).

To prevent the processes from an inadequate overgeneration, the derived forms have to be checked by an expert (linguist) mostly using some automatic tools whether these word forms occur in dictionaries or corpora. Our system allows every derived word form to be marked with a note saying that the word form is correct or that it is a potential word, not occurring in a dictionary or corpus, but it can be used. Incorrectly derived word forms are also collected as negative examples for the next iteration of the learning algorithm. We have not tried to optimise this method yet, rather, our main aim now is to implement the appropriate algorithms and representations of the lexicon so as to enable straightforward data maintenance.

Implementation of this process will be described in the section 3.3.2.

## 3. Inflectional and derivational processes

Basically, there are four major types of word-forming processes that can be distinguished:

- inflection
- derivation
- compounding
- abbreviation

Inflection refers to the systematic modification of a stem by means of suffixes and sometimes prefixes. Inflected forms express grammatical categories like case, gender or number, but do not change meaning of POS. In contrast, the process of derivation usually brings about a change in meaning and often a change in POS as well. Compounding deals with the process of merging several word bases to form a new word, whereas abbreviation shortens the word bases leaving usually just the the first letter or the first syllable.

### 3.1. Inflectional processes

Czech belongs to the family of inflectional languages which are characterised by the fact that one morpheme, typically an ending, carries the values of several grammatical categories together. For example, an ending of nouns typically expresses a value of grammatical category of case, number and gender. This feature requires a special treatment of Czech words in text processing systems. To this end, we have developed a universal morphological analyser which performs the morphological analysis based on dividing all words in Czech texts into their smallest relevant components that we call *segments*. The notion of segment roughly corresponds to the linguistic concept of *morpheme*, which denotes the smallest meaningful unit of a language.

The morphological analyser consists of three major parts: a formal description of morphological processes via morphological patterns, an assignment of Czech stems to their relevant patterns and a morphological analysis algorithm (Sedláček and Smrž, 2001).

| Case | Singular | Plural |
|---|---|---|
| Nominative | hora | hory |
| Genitive | hory | hor |
| Dative | hoře | horám |
| Accusative | horu | hory |
| Vocative | horo | hory |
| Locative | hoře | horách |
| Instrumental | horou | horami |

Table 1: The complete paradigm for the noun *hora*

The description of Czech formal morphology is represented by the system of inflectional patterns and sets of endings and it includes the lists of segments and their correct combinations. The assignment of Czech stems to their patterns is contained in the Czech Machine Dictionary (Osolsobě, 1996). Finally, the algorithm of the morphological analysis using this information splits each word into appropriate segments.

The main part of the algorithmic description of (Czech) formal morphology, as suggested in Osolsobě (1996), is a pattern definition. The basic notion is a *morphological paradigm* — a set of all forms of the inflected word where the particular forms express a system of its respective grammatical categories (see Table 1).

Data structures we decided to use for storing ending sets (e.g. S5, S2 in the following example) and inflectional patterns (e.g. ho for the noun *hora* (mountain) in this example) were described in Sedláček and Smrž (2001). The following example shows the segmentation of the word *hora* into three parts: a stem ho, an intersegment <r> and <ř> which is a standard alternation in Czech *r*→*ř* in the stem, and an ending that comes from the set S5 or S2. More precise and formal description of these structures is provided in section 3.3.

```
S5=[1FS.](a,1)(y,2)(u,4)(o,5)(ou,7)
   [1FP.](y,1)(_,2)(ám,3)(y,4)(y,5)
         (ách,6)(ami,7)
S2=[1FS.](e,3)(e,6)

ho+<r>S5|<ř>S2
```

As stated in Hajič (2000), the traditional grammars of Czech offer a much smaller paradigm system than exists in reality. For this reason we decided to build a large set of paradigm patterns to cover all the variations of Czech word forms from scratch. Fortunately, we were not limited by technical restrictions, thus we could pursue a straightforward approach within the limitations of linguistic adequacy and the robustness of the solution.

The detailed description of all variations in Czech paradigms enables us to define the application dependent generalisations of the pattern system. In its fully expanded form there are 1500 patterns. But if we do not need to take into consideration archaic word forms for a specific application, the number of paradigm patterns can be considerably reduced using automatic procedures.

### 3.2. Derivational processes

As indicated in the previous section, the morphological process of inflection is captured by means of paradigms in our system. Abbreviation and compounding do not play a crucial role in Czech morphology if compared with other languages, e. g. German (Kodydek, 2000).

The process of morphologically deriving new words, primarily with distinct POS categories, is considered to be taking place at one level higher than inflectional processes. Indeed, for example, a particular class of deverbative adjectives can be derived from the derivation paradigm of transitive verbs (see the "DEVEADJ*" relations in the following example). A hierarchical system of morphological paradigms has been implemented as a tool able to capture different levels of the Czech morphology.

Hierarchical patterns are constructed fully automatically from the binding defined on the level of basic forms always connecting one lemma with another by a specified type of a link. If a process could be described as a $n$-ary relation, it would be partitioned into $n-1$ binary relations. This partitioning is much more flexible and allows automatic generalisations of derivation relations. To demonstrate the derivation binding on the level of lemmata, we present the following example with the verb participles:

| | | |
|---|---|---|
| • DEVESUBST | | |
| *počítat (to count)* | → | *počítání (counting)* |
| • DEVEADJPAS | | |
| *počítat (to count)* | → | *počítaný (counted)* |
| • DEVEADJPASSHORT | | |
| *počítat (to count)* | → | *počítán (is counted)* |
| • DEVEADJACTIMPF | | |
| *počítat (to count)* | → | *počítající (is counting)* |

From this example it follows that each link connects one base form of a word with another one and names such relation. If the label of a base form is unambiguous and can therefore can be used as a primary identifier, it is sufficient to specify only these labels in the binding process. If the label itself can be ambiguous, the pairs of lemma and the relevant inflectional pattern are connected. However, even this approach is not able to completely represent the dependency of the relation on a particular sense of a word. For example, the following relation between a noun and the derived possessive adjective

| | | |
|---|---|---|
| • POSSADJ | | |
| *editor (editor)* | → | *editorův (editor's)* |

is valid only for the reading *editor* denoting a man who edits books but not for the reading that denotes a computer program, the second meaning of the noun *editor*. This is why we have implemented a system connecting the triplet pairs, namely sense-id, lemma and paradigm, by a named relation.

The indexing techniques and dictionary methods (Knuth, 1973) used in our implementation allow an efficient retrieval of related lemmata. It is also possible to quickly return a chosen base form for a set of related words – a feature which is much preferred in some applications, e.g. in the area of information retrieval or indexing Internet documents.

The system of base form binding is not limited to the basic derivative processes described above. The same principle, for example, depicts two types of relation on the level under the basic derivation, namely original/adapted orthography and inflectional/non-inflectional doublets in the case of the loanwords.The former can be demonstrated by the example of a link between *gymnasium/gymnázium (high school)* (in the actual version of our morphological analyser we use an even more elaborated assignment of these doublet types in the form of a basic type of relation and more specific subtype). The example of an inflectional/non-inflectional doublet is the link between the word *abbé* assigned to the paradigm *abbé* (non-inflectional) and *Tony*. It is of course possible to model such relations on the basic level of inflectional paradigms as a word-form homonymy. However, it would lead to the mixture of unrelated forms and would complicate special types of analyses, e.g. a style-checker analysis could make very interesting findings.

There are other relations connecting lemmata above the level of basic derivative processes. We take advantage of the standard process and are able to uniformly describe such different relations as diminutives (and their degrees):

| | | |
|---|---|---|
| • DIMIN:1 | | |
| *vůz (wagon)* | → | *vozík (little wagon)* |
| • DIMIN:2 | | |
| *vůz (wagon)* | → | *vozíček (very little wagon)* |

aspectual relations of verbs:

| | | |
|---|---|---|
| • ASPPAIR | | |
| *říci (to say)* | → | *říkat (be saying)* |

iterative relations of the verbs (together with "*degrees*"):

| | | |
|---|---|---|
| • ITER:1 | | |
| *chodit (to go)* | → | *chodívat (go regularly)* |
| • ITER:2 | | |
| *chodit (to go)* | → | *chodívávat (used to go)* |

the relation between an animate noun and a derived possessive adjective:

| | | |
|---|---|---|
| • MASCPOSS | | |
| *otec (father)* | → | *otcův (father's)* |

the process of converting masculine nouns to feminines:

| | | |
|---|---|---|
| • MASC2FEMI | | |
| *soudce (judge)* | → | *soudkyně (judge female)* |

or synonyms and antonyms:

| | | |
|---|---|---|
| • SYNO | | |
| *kosmonaut (cosmonaut)* | → | *astronaut (astronaut)* |
| • ANTO | | |
| *mladý (young)* | → | *starý (old)* |

The last class of the links brings us directly to other relations that can be found in semantic networks like Wordnet (Miller, 1993). The typical relations of hyperonymy/hyponymy, meronymy (part/whole) etc. are modelled on the higher level, the level based on synonyms, so that groups of synonyms ("*synsets*" as they are known in Wordnet) can be linked.

The possibility of building complex structures of links, e.g. relations of relations, is also employed in connecting roots of loanwords to their Czech equivalents. We are therefore able, like Páleš (1994), to relate words derived from the Greek root *kard* with the group of Czech words derived from the Czech root *srd* (heart), e.g. *osrdečník* (pericardium), *kardiostimulátor* (pacemaker), *srdce* (heart), *kardiologie* (cardiology).

### 3.3. Formal description of the derivational processes

Formally we can define all the potential sets of patterns:

$\mathcal{T}$ – set of all possible tags
$\mathcal{A}$ – alphabet, $\mathcal{A}^*$ – set of all chains on alphabet $\mathcal{A}$
$\mathcal{P}$ – class containing all sets of patterns can be defined recursively:

$$\begin{aligned} &1)\ \emptyset \in \mathcal{P} \\ &2)\ P \in \mathcal{P} \Rightarrow Q \cup P \in \mathcal{P}, \text{ where} \\ &\quad Q = \{(p, \overline{p}, s, \overline{s}, t, L) | p, \overline{p}, s, \overline{s} \in \mathcal{A}^* \\ &\qquad \wedge t \subseteq \mathcal{T} \wedge L \subseteq P\} \\ &3)\ \text{nothing else is a set of patterns} \end{aligned} \tag{1}$$

$S_x(l) : \mathcal{A}^* \rightarrow \mathcal{A}^*$ – simple (one step) substitution for the pattern $x \equiv (p, \overline{p}, s, \overline{s}, t, L) \in \mathcal{P}$ and word form $l \equiv \overline{p} \oplus r \oplus \overline{s}$, where $r \in \mathcal{A}^*$ , is defined:

$$S_{(p,\overline{p},s,\overline{s},t,L)}(\overline{p} \oplus r \oplus \overline{s}) = p \oplus r \oplus s \tag{2}$$

$T(l, x) : \mathcal{A}^* \times \mathcal{P} \rightarrow \wp(\mathcal{A}^* \times \wp(\mathcal{T}))$ – function for derivation of tagged word forms from $l \in \mathcal{A}^*$ using pattern $x \equiv (p, \overline{p}, s, \overline{s}, t, L) \in \mathcal{P}$ can be explained as:

$$\begin{aligned} &T(l, (p, \overline{p}, s, \overline{s}, t, L)) \equiv K, \text{ where} \\ &\text{if } L = \emptyset \text{ then } K = \{(S_x(l), t)\} \\ &\text{else } \quad K = \{(k, t \cup R) | y \in L \\ &\qquad \wedge (k, R) \in T(S_x(l), y)\} \end{aligned} \tag{3}$$

Introduced formalism is partially demonstrated in Table 2. This example describes word forming process, which derive word forms from family names (part of one pattern). The family names in Czech can be divided to three classes based on:

- male name (MAL)
- female name (FEM)
- name labelling all family together (FAM)

The example shows that all three groups of word forms can be divided grammatically into substantive (SUB), possessive (POS) adjective (ADJ).

The groups of possessive adjectives can be separated[1] by the grammatical gender of an object which is possessive into male, female or to the family. The clusters can be further refined according to grammatical number, ....

Our approach is very similar to "*two level morphology*" described in Koskenniemi (1983). We introduce a hierarchical model of patterns which enables us to separate the sets of patterns for inflectional and derivational processes. The patterns can be named, marked according to their function, and sorted by linguists. A finely-tuned database of the

[1] not show in the example for a simplicity's sake

patterns will be used as a formal description of word forming processes in natural languages.

Our database starts with the inflectional level only and we will explain the process of adding a derivation level. Generally we can start with patterns (Goldsmith's "*signatures*") given from the unsupervised learning of corpora data, see Goldsmith (2001).

In our application of this work, we will not use all elements of set $\mathcal{P}$. One finite set $\mathcal{Q}$ will be selected for the specified language at one point in time:

$$\mathcal{Q} \in \mathcal{P} \wedge \forall(p, \overline{p}, s, \overline{s}, t, L) \in \mathcal{Q} \Rightarrow L \subset \mathcal{Q} \tag{4}$$

The stem dictionary $\mathcal{S}$ is a set of tuples $(l, v)$, where $l \ldots$ lemma, $v \ldots$ pattern:

$$\mathcal{S}^{\mathcal{Q}} = \{(l, v) | l \in \mathcal{A}^* \wedge v \in \mathcal{Q}\} \tag{5}$$

The whole dictionary $\mathcal{D}_{\mathcal{S}^{\mathcal{Q}}}$ for the stem dictionary $\mathcal{S}$ (see equation 4) and the set of patterns $\mathcal{Q}$ (see equation 5) will be defined:

$$\mathcal{D}_{\mathcal{S}^{\mathcal{Q}}} = \bigcup_{(l,v)\in\mathcal{S}} T(l, v) \tag{6}$$

The applications which use this formal description require that the stem be determined from the lemma and from the pattern: the reverse is also possible. We will next introduce set $\mathcal{L}$[2].

There is one to one correspondence between the pattern $v$ and the triple $(\overline{p}, \overline{s}, t)$, where the $\overline{p} \ldots$ the prefix, $\overline{s} \ldots$ the suffix, and $t \ldots$ the set of tags. Let's assign $w \equiv (\epsilon, \overline{p}, \epsilon, \overline{s}, t, \emptyset)$ and $z \equiv (\overline{p}, \epsilon, \overline{s}, \epsilon, t, \emptyset)$, than:

$$\begin{gathered}\mathcal{L} = \{(v, w, z) | v \in \mathcal{Q} \wedge \overline{p}, \overline{s} \in \mathcal{A}^* \wedge t \subseteq \mathcal{T}\} \wedge \\ \wedge \mathcal{Q} \subseteq \{v | \exists (v, w, z) \in \mathcal{L}\} \wedge \\ \wedge (v, w_1, z_1), (v, w_2, z_2) \in \mathcal{L} \Rightarrow (w_1, z_1) = (w_2, z_2)\end{gathered} \tag{7}$$

We define the lemmatisation function $lm$, which transforms the stem (or root) $r$ assigned to the pattern $v$ to the tagged base form:

$$\begin{gathered}lm(r, v) = (S_z(k), t); \exists (v, w, z) \in \mathcal{L}, \text{where} \\ z \equiv (\overline{p}, \epsilon, \overline{s}, \epsilon, t, \emptyset)\end{gathered} \tag{8}$$

It is possible to define function $st$ "*inversive*" to function $lm$, transforming base form $l$ assigned to pattern $v$ to relevant stem.

$$st(l, v) = S_w(l); \exists (v, w, z) \in \mathcal{L} \tag{9}$$

### 3.3.1. Word-clusters

A set of word forms derived using a pattern from one lemma (stem, root) will be called a *cluster*. Word forms can be sorted hierarchically within a cluster. The hierarchical structure is based on breaking it down into equivalence classes. The components of a such decomposition (clusters — equivalence classes) will be called *sub-clusters*. Those sub-clusters can be divided into other equivalence classes. The hierarchy of word clusters is based on a hierarchical model of patterns introduced at the top of section 3.3. Subclusters are refined according to the structure of the associated pattern $(p, \overline{p}, s, \overline{s}, t, L)$ and the set of "*sub-patterns*" $L$.

[2]members are triples of patterns

As can be seen in Table 2, such a system enables the amount of duplicated information in our morphological database to be reduced. In this example we demonstrate that we can define the same dictionary (see equation 6) from different stem dictionaries (see equation 5):

$$\begin{aligned}\mathcal{S}_1^{\mathcal{Q}} &= \{(\textit{Aleš}, \texttt{Aleš\_F})\} \\ \mathcal{S}_2^{\mathcal{Q}} &= \{(\textit{Aleš}, \texttt{Aleš\_P}), (\textit{Alšová}, \texttt{Alšová\_P}), \\ &\quad (\textit{Alšovi}, \texttt{Alšovi\_P})\} \\ \mathcal{S}_3^{\mathcal{Q}} &= \{(\textit{Aleš}, \texttt{Aleš}), (\textit{Alešův}, \texttt{otcův}), \\ &\quad (\textit{Alšová}, \texttt{Alšová}), (\textit{Alšové}, \texttt{Alšové}), \\ &\quad (\textit{Alšovi}, \texttt{Novákovi}), \\ &\quad (\textit{Alšových}, \texttt{Novákových})\}\end{aligned}$$

$$\mathcal{D}_{\mathcal{S}_1^{\mathcal{Q}}} \doteq \mathcal{D}_{\mathcal{S}_2^{\mathcal{Q}}} \doteq \mathcal{D}_{\mathcal{S}_3^{\mathcal{Q}}}$$

where $\doteq$ is equivalence only for the word forms, ignoring adding of some tags in some derivational levels. Assign $w \equiv (p, \overline{p}, s, \overline{s}, t, L)$, when $t = \emptyset$ for any used $w$, than it is real equivalence because of:

$$T(l, w) = \bigcup_{y\in L} T(S_w(l), y) \tag{10}$$

A potential application of our approach is the construction of a stem(root)-dictionary.

### 3.3.2. Dictionary extension

The user view of this process is described in Section 2. The underlying mechanism works like this: as was discussed in Section 3.3.1., the same dictionary can be based on different stem dictionaries. We are searching for some low level morphological patterns which can be connected by some higher level derivational pattern. For this search, the examples gathered by the user (see section 2.) can be used. Examples are n-tuples of members of the stem dictionary (see equation 5)

$$e = (m_1, \ldots, m_n); \text{ where } m_1, \ldots, m_n \in \mathcal{S}^{\mathcal{Q}}$$

Appropriate patterns that connect low level ones by transforming their base forms are suggested to the user. The words assigned to those low level sub-patterns are derived from the lemmas $hl$ by suggested higher level patterns $hp$.

$$\begin{aligned}&hl = \overline{p} \oplus r \oplus \overline{s} \\ &hp = (\epsilon, \overline{p}, \epsilon, \overline{s}, \emptyset, L), \text{where} \\ &L = \{(p_i, \epsilon, s_i, \epsilon, \emptyset, \{v_i\}) | (p_i \oplus r \oplus s_i, v_i) = m_i\}\end{aligned}$$

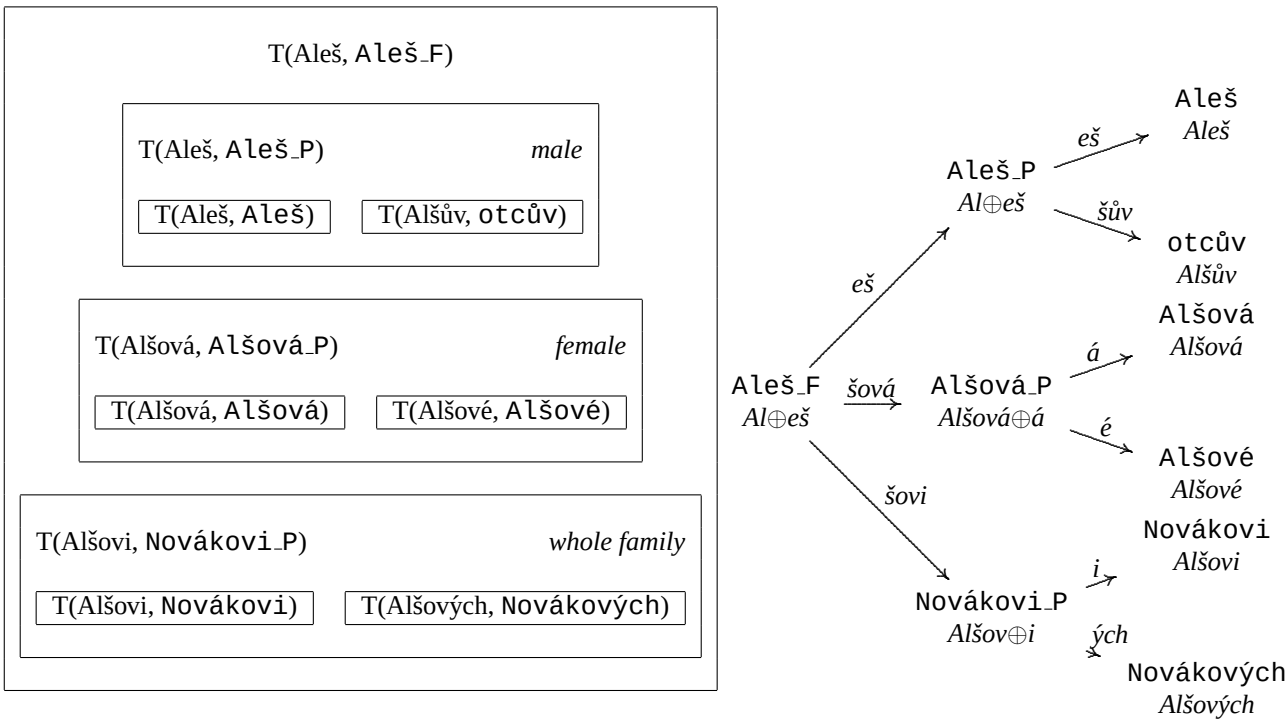
The stem (root) $r$ is computed for any example of $e$ as the longest common substring of the strings $p_i \oplus r \oplus s_i$, where $(p_i \oplus r \oplus s_i, v_i) = m_i$. The values of $p_i$ and $s_i$ are counted in the next step. The lemma $hl$ can be selected as one of the members $p_i \oplus r \oplus s_i$, then we assign $\overline{p} = p_i$ and $\overline{s} = s_i$.

Now we can reduce the stem dictionary size for all given examples, replacing all members of $e$ by $(hl, hp)$:

$$\mathcal{S}_2^{\mathcal{Q}\cup\{hp\}\cup L} = \mathcal{S}_1^{\mathcal{Q}} \setminus \{m | m \in e\} \cup \{(hl, hp)\}$$

But this is the reduction over the stem dictionary for the manually assigned example only. When we want to

$\mathcal{T} = \{\text{MAL, FEM, FAM, SUB, ADJ, POS}\}$
$\mathcal{Q} = \{\texttt{Aleš\_F}, \texttt{Aleš\_P}, \texttt{Alešová\_P}, \texttt{Novákovi\_P}, \ldots\} \in \mathcal{P}$
$\texttt{Aleš\_F} \equiv (\epsilon, \textit{eš}, \epsilon, \epsilon, \emptyset, \{(\textit{eš}, \epsilon, \epsilon, \epsilon, \emptyset, \{\texttt{Aleš\_P}\}), (\textit{šová}, \epsilon, \epsilon, \epsilon, \emptyset, \{\texttt{Alšová\_P}\}), (\textit{šovi}, \epsilon, \epsilon, \epsilon, \emptyset, \{\texttt{Novákovi\_P}\})\})$
$\texttt{Aleš\_P} \equiv (\epsilon, \textit{eš}, \epsilon, \epsilon, \{\text{MAL}\}, \{(\textit{eš}, \epsilon, \epsilon, \epsilon, \{\text{SUB}\}, \{\texttt{Aleš}\}), (\textit{ův}, \epsilon, \epsilon, \epsilon, \{\text{POS,ADJ}\}, \{\texttt{otcův}\})\})$
$\texttt{Alšová\_P} \equiv (\epsilon, \textit{á}, \epsilon, \epsilon, \{\text{FEM}\}, \{(\textit{á}, \epsilon, \epsilon, \epsilon, \{\text{SUB}\}, \{\texttt{Alšová}\}), (\textit{é}, \epsilon, \epsilon, \epsilon, \{\text{POS,ADJ}\}, \{\texttt{Alšové}\})\})$
$\texttt{Novákovi\_P} \equiv (\epsilon, i, \epsilon, \epsilon, \{\text{FAM}\}, \{(i, \epsilon, \epsilon, \epsilon, \{\text{SUB}\}, \{\texttt{Novákovi}\}), (\textit{ých}, \epsilon, \epsilon, \epsilon, \{\text{POS,ADJ}\}, \{\texttt{Novákových}\})\})$




$$T(\text{Aleš}, \texttt{Aleš\_F}) \doteq T(\text{Aleš}, \texttt{Aleš\_P}) \cup T(\text{Alšová}, \texttt{Alšová\_P}) \cup T(\text{Alšovi}, \texttt{Novákovi\_P})$$
$$T(\text{Aleš}, \texttt{Aleš\_P}) \doteq T(\text{Aleš}, \texttt{Aleš}) \cup T(\text{Alšův}, \texttt{otcův})$$
$$T(\text{Alšová}, \texttt{Alšová\_P}) \doteq T(\text{Alšová}, \texttt{Alšová}) \cup T(\text{Alšové}, \texttt{Novákové})$$
$$T(\text{Alšovi}, \texttt{Novákovi\_P}) \doteq T(\text{Alšovi}, \texttt{Novákovi}) \cup T(\text{Alšových}, \texttt{Novákových})$$
$$T(\text{Alšovi}, \texttt{Novákovi\_P}) = \{(k, t \cup \{\text{SUBS,FAM}\}) | (k,t) \in T(\text{Alšovi}, \texttt{Novákovi})\} \cup \{(k, t \cup \{\text{POS,ADJ,FAM}\}) | (k,t) \in T(\text{Alšových}, \texttt{Novákových})\}$$

Table 2: Example of word cluster hierarchy of the family name *Aleš*

extend our dictionary we can try to check if the words (from set $E(v_i)$) are assigned to any pattern $v_i$, where:

$$E(v_i) = \{l | (l, v_i) \in \mathcal{S}^{\mathcal{Q}}\}$$

can be reduced by a higher level pattern.

The sets of potential "*high level*" lemmas are as follows:

$$hl_i = \{\overline{p} \oplus r \oplus \overline{s} | x \in E(v_i) \wedge x = p_i \oplus r \oplus s_i\}$$

we know that:

$$T(p_i \oplus r \oplus s_i, v_i) \subset T(\overline{p} \oplus r \oplus \overline{s}, hp) \text{ for any } r \in \mathcal{A}^* \quad (11)$$

The derived word forms can be checked if they are assigned to the other appropriate connected patterns. There are two cases:

1. $hl \in \bigcap_{\forall i} hl_i$ — can be simply reduced (resulted from equation 11)
2. $hl \in \bigcup_{\forall i} hl_i \setminus \bigcap_{\forall i} hl_i$ — some words, automatically computed from $hl_i$, can be added, thereby extending the dictionary, and then it can be reduced.

## 4. Implementation

The word derivation process in Czech consists of several stages. Each step can be realized as a module understood as a procedure starting with a basic word form on the input and producing a tuple of derived word forms in the output. More precisely, in our system the module can be defined by using n+1 strings and implemented as a simple non-deterministic finite-state translation automaton without loops. The automaton substitutes the string $s_0$ (the suffix of an input lemma) with the strings $s_1 \ldots s_n$ for each output tuple members $(t_1, \ldots, t_n)$. The same can be done with prefixes ($p_0 \rightarrow p_1 \ldots p_n$) and some tags can be assigned to the $t_i$ members. This automaton can be constructed in accordance with the introduced formal description using equations 5, 4, 3.

| | |
|---|---|
| #intersegments | 779 |
| #endings | 643 |
| #sets of endings | 2,806 |
| #patterns | 1,570 |
| #stem bases | 223,600 |
| #generated word forms | 5,678,122 |
| #generated tags | 1,604 |
| speed of the analysis | 20,000 words/s |
| dictionary | 1,930,529 Bytes |
| morphological information | 147,675 Bytes |

Table 3: Statistical data

| **suffix** | **freq.** | **suffix** | **freq.** | **suffix** | **freq.** |
|---|---|---|---|---|---|
| *-tel* | 911 | *-dlo* | 483 | *-ní* | 30454 |
| *-ce* | 232 | *-tko* | 379 | *-tí* | 3622 |
| *-ař* | 456 | *-an* | 254 | *-ost* | 7871 |
| *-ář* | 916 | *-anka* | 218 | *-ota* | 211 |
| *-ista* | 904 | *-iště* | 377 | *-ství* | 2081 |
| *-íček* | 733 | *-árna* | 360 | *-ctví* | 820 |
| *-ička* | 1545 | *-ovna* | 208 | *-stvo* | 270 |
| *-i/íčko* | 391 | *-yně* | 318 | *-ctvo* | 141 |
| **Total: 54155** | | | | | |

Table 4: Frequency of selected suffixes in noun lemmata

Modules can be used in cascades as allowed by recursion in equation 3, i.e. a derived output word form from one module can be used as an input lemma for the next one and thus a tree-based hierarchy in the derivation process can be created. Some lemma (with its hierarchy sorted according to its derivation process) can be shown in the output using the set $\mathcal{L}$, see equation 7.

The key point of the successful implementation of the analyser is an efficient storage mechanism for lexical items. A *trie* structure is used for storing stem bases of Czech word forms. One of the main disadvantages of this approach are high memory requirements. We tried to solve this problem by implementing the trie structure in the form of the minimal finite state automaton. The incremental method of building such an automaton was presented in Daciuk et al. (1998) and is fast enough for our purpose. Moreover, the memory requirements for storing the minimal automaton are significantly lower (see Table 3).

The power of the analyser can be evaluated by two features. The most important one is the number of words that can be recognised by the analyser. This number depends on the quality and richness of the dictionary. Our database contains 223,600 stem bases and `ajka` is able to analyse and generate 5,678,122 correct Czech word forms. The second feature is the speed of analysis. In the brief mode, `ajka` can analyse more than 20,000 words per second on a PentiumIII processor with a frequency of 800MHz. Some other statistical data, such as the number of segments and size of binary files, is shown in the following Table 3.

## 5. Conclusions

This approach has been implemented as the morphological tool, I_PAR, (Veber, 2001) (which will be demonstrated). The results of the experiments with I_PAR will be presented in the paper. They capture 5 agentive suffixes (*tel, ce, ař, ář, ista*), 3 diminutive suffixes (*íček, ička, i/íčko*), 2 instrument suffixes (*dlo, tko*), 2 inhabitant suffixes (*an, anka*), 3 location suffixes (*iště, árna, ovna*), 1 opposite-sex suffix (*yně*), 2 action suffixes (*ní, tí*), 4 property suffixes (*ost, ota, ství, ctví*) and 2 collective suffixes (*stvo, ctvo*); see Table 4. The procedure and its results is going to be used to mark the selected ILR in the Czech WordNet, which will be done within the EU Project Balkanet (Bal, 2001) and possibly extended to Bulgarian and Serbo-Croatian).

## 6. References


2001. Balkanet project. http://dblab.upatras.gr.

J. Daciuk, R. E. Watson, and B. W. Watson. 1998. Incremental Construction of Acyclic Finite-State Automata and Transducers. In *Finite State Methods in NLP*, Bilkent University, Ankara, Turkey, June – July.

J. Goldsmith. 2001. Unsupervised Learning of the Morphology of a natural Language. Department of Linguistics University of Chicago.

J. Hajič. 2000. *Disambiguation of Rich Inflection*. Charles University Press, 1st edition.

J. Klímová and K. Pala. 2000. Application of WordNet ILR in Czech Word-formation. In M. Gavrilidou, editor, *Proceedings of LREC 2000*, pages 987–992.

D. E. Knuth. 1973. *The Art of Computer Programming: Sorting and Searching*, volume 3. Addison Wesley, 2nd edition.

G. Kodydek. 2000. A Word Analysis System for German Hyphenation, Full Text Search, and Spell Checking .... In P. Sojka, I. Kopeček, and K. Pala, editors, *Proceedings of TSD 2000*, pages 39–44, Brno, Czech Republic, Sep. Springer-Verlag.

K. Koskenniemi. 1983. Two-level morphology: A general computational model for word-form recognition and production. Technical report, University of Helsinki, Department of General Linguistics.

G. A. Miller. 1993. Five papers on Wordnet. Technical report, Princeton.

K. Osolsobě. 1996. *Algorithmic Description of Czech Formal Morphology and Czech Machine Dictionary*. Ph.D. thesis, Faculty of Arts, Masaryk University Brno. In Czech.

E. Páleš. 1994. *Sapfo - Paraphraser of Slovak*. Veda, Bratislava.

R. Sedláček and P. Smrž. 2001. A New Czech Morphological Analyser `ajka`. In *Proceedings of TSD 2001*, pages 100–107, Berlin. Springer-Verlag.

M. Veber. 2001. *Tools for Text Corpora and Morphological Databases*. Ph.D. thesis, Faculty of Informatics, MU, Brno. Dissertation Thesis, manuscript, in Czech.