Jan HAJIČ Research Institute of Mathematical Machines Loretánské nám. 3 118 55 Praha 1, Czechoslovakia

### Abstract

A formalism for the description of a system of formal morphology for flexive and agglutinative languages (such as Czech) is presented, borrowing some notions and the style from the theory of formal languages. Some examples (for Czech adjectives) are presented at the end of the paper. In these examples, the formalism's rules are used for the phonology-based changes as well, but nothing prevents the use of a separate phonology level (e.g. of the Koskenniemi's two-level model) as a front- (and back-) end for the analysis (and synthesis).

#### 1. The Motivation

Using a computer, the morphological level is a basis for building the syntacticosemantic part of any NL analysis. The CL world pays more attention to morphology only after the work /Koskenniemi 1983/ was published. However, as Kay remarked (e.g. in /Kay 1987/), phonology was actually what was done in /Koskenniemi 1983/. Moreover, the strategy used there is best suited for agglutinative languages with almost one-toone mapping between morpheme and grammatical meaning, but slavonic languages are different in this respect.

One of the practical reasons for formalizing morphology is that although there are some computer implementations using a Czech morphology subsystem (/Hajič,Oliva 1986/, /Kirschner 1983/, /Kirschner 1987/), based on the same sources (/EBSAT VI 1981/, /EBSAT VII 1982/), no unifying formalism for a complete description of formal morphology exists.

#### 2. The Formalism

The terms alphabet, string, concatenation, ..., symbol N (positive integers), indexes \* and \* are used here in the same way as in the formal grammar theory; the symbol exp(A) denotes the set of all subsets of A, e denotes an empty string. Uppercase letters are used mainly for denoting sets and newly defined structures, lowercase letters are used for mappings, for elements of an alphabet and for strings.

**Definition 1.** A finite set K of symbols is called a set of grammatical meanings (or simply meanings for short); values from K represent values of morphological categories (e.g. sg may represent singular number, p3 may represent dative ("3rd case") for nouns, etc.).

**Definition 2.** A finite set  $D = ((w, i) \in A^* x (N \cup \{0\}))$ , where A is an alphabet, is called a dictionary. A pair (w,i)  $\in D$  is called a dictionary entry, w is a lexical unit and i is called pattern number. In the linguistic interpretation, a lexical unit represents the

notion "systemic word", but it need not be represented by a traditional dictionary form.

**Definition 3.** Let A be a finite alphabet, K a finite set of meanings, V a finite alphabet of variables such that A  $\wedge$  V = {}. The quintuple (A,V,K,t,R), where t is a mapping t: V -> exp(A\*) assigning types to variables, R is a finite set of rules (I,H,u,v,C), where I  $\wedge$  N is is a finite set (of labels), C  $\wedge$  (N  $\vee$  (O)) is a finite set (of continuations), H  $\wedge$  K is a set of meanings belonging to a particular rule from R, u,V  $\in$  (A  $\vee$  V)<sup>\*</sup>, is called a controlled rewriting system (CRS); all variables from the left-hand side (u) must be present on the right-hand side (v) and vice versa (rule symmetry according to variables).

**Definition 4.** Let T = (A, V, K, t, R) be a CRS. A (simple) substitution on T will be any mapping q:  $V \rightarrow A^*$ ;  $q(v) \in t(v)$ .

**Definition 5.** Let T = (A, V, K, t, R) be a CRS and q a simple substitution on T. Mapping d:  $(A \lor V)^* \to A^*$  such that d(e) = e; d(a) = afor  $a \in A$ ; d(v) = q(v) for  $v \in V$ ; d(bu) =d(b)d(u) for  $b \in (A \lor V)$ ,  $u \in (A \lor V)^*$  will be called (generalized) substitution derived from q.

<u>Comment.</u> The (generalized) substitution substitutes (in a given string) all variables by some string. The same string is substituted for all occurences of this variable (follows from the definition).

**Definition 6.** Let T = (A, V, K, t, R) be a CRS and F A K. Let then G, G' A K, w,z  $\in$  (A v V)<sup>\*</sup>, i  $\in$  N, i'  $\in$  (N v (O)). We say that w can be directly rewritten in the state (i,G) to z with a continuation (i',G') according to meanings F (written as w(i,G)  $\Rightarrow$ )[T,F] z(i',G')), if there exist such rule (I,H,u,v,C)  $\in$  R and such simple substitution q on T, that i  $\in$  I, i'  $\in$  C, H A F, G = G' v H, d(u) = w and d(v) = z, where d is the substitution derived from q. Relation =><sup>\*</sup>[T,F] is defined as the reflexive

Relation = \*[T,P] is defined as the reflexive and transitive closure of = [T,P].

<u>Comment.</u> The CRS is controlled through continuations and labels. After a direct rewriting operation, the only rules that could be applied next must have in their label at least one number from the rewriting operation continuation. Please notice that: - this operation always rewrites whole words;

- this operation analysis remarks and righthand side of a rule that it should be only string (of letters and/or variables) is not so strong as it may seem, because no restrictions are imposed on the substitution q. However, to be able to implement the rules in a particular implementation as finite state machines, we shall require q to be defined using regular expressions only.

**Definition 7.** Let  $T = (A, V, K, \tau, R)$  be a CRS and let n be the maximal number from all

labels from all rules from R; n-tuple P = (p1, ..., pn) will be called a list of patterns on T (the elements of P are called patterns) if for every i a mapping pi: exp(K)x A\* ->  $\exp(A^*)$  is defined as z e pi(P,w) <=w(i,F)  $=>^{*}[T,F] z(0,()).$ 

Comment. The "strange" sets G and G' from the definition 6 acquire a real meaning only in connection with the definition of patterns; they have a controlling task during pi construction, namely, they check whether all meanings from F are used during the derivation. "To use a meaning k" means here that there is some rule (I,H,u,v,C) applied in the course of derivation from w(i,F) to  $z(0, \{\})$  such that k  $\in$  H. Such meaning can then be removed from G when constructing G' (see Def. 7); meanings not from H cannot. Thus, to get the empty set in z(0,()) when starting from w(i,F), all meanings from F must be "used" in this sense.

A pattern describes how to construct to a given word w all possible forms according to meanings F. In this sense, the notion of pattern does not differ substantially from the traditional notion of pattern in formal morphology, although traditionally, not the constructive description, but just some representative of such a description is called a pattern.

**Definition 8.** Let D be a dictionary over an alphabet A, T = (A, V, K, t, R) a CRS and P a list of patterns on T. A quadruple M =(A,D,K,P) is called a morphology description on T (M['(]-description).

**Definition 9.** Let T = (A, V, K, t, R) be a CRS **Definition** 7. Let i = (A, V, K, t, K) be a CRS and M = (A, D, K, P) an MITJ-description. Set L =  $(z \in A^{X};$  there ex.  $w \in A^{X}$ ,  $i \in N$ , H  $\land K; z$ E pi(H, w)) will be called a language generated by MITJ-description M. The elements of L will be called word forms. Comment. The term morphology description introduced above is a counterpart to a

description of a system of formal morphology, as used in traditional literature on morphology.

Definition 9 is introduced here just for the purpose of formalization of the notion of word form, i.e. any form derived from any word from the dictionary using all possible meanings according to M[T].

**Definition 10.** Let T = (A, V, K, t, R) be a CRS and M = (A,D,K,P) be MIT]-description. The and n = (A, D, K, P) be nerr-description. The term synthesis on M is used for a mapping s:  $\exp(K) \ge A^* \rightarrow \exp(A^*)$ ;  $s(H, w) = \{z; ex. i \in$ N, i  $\langle = n; z \in pi(H, w) \& (w, i) \in D$ . The term analysis is used then for a mapping a:  $A^* \rightarrow \exp(\exp(K) \ge A^*)$ ;  $a(z) = ((H, w); z \in$ s(H,w)).

According to definition 10. Comment. synthesis means to use patterns for words from the dictionary only. The definition of analysis is based on the synthesis definition, so it clearly and surely follows the intuition what an analysis is. In this sense, these definitions don't differ substantially from the traditional view on formal morphology, as opposed to Koskenniemi; however, the so-called complex word forms ("have been called") are not covered, and their analysis is shifted to syntax.

The definition of analysis is quite clear, but there is no procedure contained, capable of actually carrying out this process. However, thanks to rule symmetry it is possible to reverse the rewriting process:

**Definition 11.** Let T = (A, V, K, t, R) be a CRS. Further, let G, G'  $\land K$ , i  $\in N$ , i'  $\in (N \lor (0))$ ,  $z, w \in A^{\pm}$ . We say that under the condition (i',G') it is possible to directly analyse a string z to w with a continuation (i,G) (we write  $z(i',G') \approx (T] w(i,G)$ ), if there exists a rule  $(I,H,u,v,C) \in \mathbb{R}$  and a simple substitution q on T such that  $i \in I$ , i'  $\in$  C, G = G'  $\lor$  H, d(u) = w a d(v) = z, where d is the generalized substitution derived from q. A relation "it is possible to analyze" (= $\langle *[T] \rangle$ ) is defined as a reflexive and transitive closure of =<[T].

**Definition 12.** Let T = (A, V, K, t, R) be a CRS and  $z \in A^*$ . Every string  $w \in A^*$ ,  $i \in N$  and F $\land K$  such that  $z(O, (...) = \langle T, T \rangle w(i, F)$  is called a predecessor of z with a continuation (i,F). Lemma. Let T = (A, V, K, t, R) be a CRS and w  $\in A^*$  a predecessor of string  $z \in A^*$  with a continuation (i, F). Then  $z \in pi(F, w)$ , where pi is a pattern by T (see Def. 7). Proof (idea). The only "asymmetry" in the definition of => as opposed to =<, i.e. the condition  $H \wedge P$ , can be solved putting (see Def. 11)  $P = () \lor H1 \lor H2 \lor \cdots \lor Hn$  (for n analysis steps). Then surely Hi n F for every i.

**Theorem.** Let T = (A, V, K, t, R) be a CRS. M = (A,D,K,P) an MITI-description, a an analysis by M and w  $\in A^*$  a predecessor of  $z \in A^*$  with a continuation (i,F). Moreover, let (w,i) e D. Then  $(F,w) \in a(z)$ .

Proof follows from the preceding lemma and from the definition of analysis.

Comment. This theorem helps us to manage an analysis of a word form: we begin with the form being analysed (z) and a "continuation" (0,()), using then "reversed" rules for back rewriting. In any state w(i,F) during this process, a correct analysis is obtained whenever (w,i) is found in the dictionary. At the same time we have in F the appropriate meanings. Passing along all possible paths of back rewriting, we obtain the whole set a(z).

#### 3. An Example

illustrate the To most important features of the formalism described above, we have chosen a simplified example of Czech adjectives (regular declination according to two traditional "patterns" - mladý (young) and jarní (spring), with negation, full comparative and superlative, sg and pl, but only masc. anim. nominative and genitive).

The dictionary:

D = {(nový,1), new
 (podlý,2)} vile (it has no neg. forms)

The CRS: CRS T = (A,V,K,t,R):  $A = \{a, \hat{a}, \hat{b}, c, \check{c}, \dots, z, \check{z}, \#\}$ (# means word separator) K = {sg,pl,comp,sup,neg,masc,nom,acc}  $V = \{-, L_{1}M\}$ t(-) = A<sup>\*</sup>; t(L) = {1,z}; t(M) = {m,n,v} R = { (see fig. 1) }

((1), ( ), -, -, (2)), ((3), (masc, sg, nom), - $\dot{y}$ , - $\dot{y}$ #, (0)), ((1), (neg ), -, ne~, (2)), ((3), (masc, sg, acc), - $\dot{y}$ , -ého#, (0)), ((2), ( ), -, -, (3)), ((3), (masc, pl, nom), - $\dot{y}$ , -i#, (0)), ((2), (comp), -L $\dot{y}$ , -Lejší, (3)), ((3), (masc, pl, acc), - $\dot{y}$ , -é#, (0)), ((2), (sup ), -L $\dot{y}$ , nej-Lejší, (3)), ((3), (masc, sg, nom), -i, -i#, (0)), ((2), (comp), -M $\dot{y}$ , -M $\dot{z}$ jší, (3)), ((3), (masc, sg, acc), -i, -ihu#, (0)), ((2), (comp), -M $\dot{y}$ , -M $\dot{z}$ jší, (3)), ((3), (masc, sg, acc), -i, -ihu#, (0)), ((2), (comp), -M $\dot{y}$ , -M $\dot{z}$ jší, (3)), ((3), (masc, sg, acc), -i, -ihu#, (0)), ({2},{sup },-Mý,nej-Mější,{3}), ({3},{masc,pl,nom}, -í, -í#,{0}), ({3},{masc,pl,acc}, -i, -i#,{0}) Fig. 1 using p2: podlý(2,{sup,masc,p1,acc}) => two possib. nejpodlejši(3,{masc,pl,acc}) => ist alt. podly(3,{sup,masc,pl,acc}) => 2nd alt. this is not a solution Possibilities without removing "used" meanings are not shown; all lead to non-empty G' in the resulting z(0,G'). Fig. 2 nejnovější#(0,{}) =< nejnovějši(3,{masc,pl,acc}) =< ....not in D (4 alter.)
nový(2,{sup,masc,pl,acc}) =< ....not in D
nový(1,{sup,masc,pl,acc}).....e D; SOLUTION
nejnový(2,{comp,masc,pl,acc}) =< not in D (2 alter.)
jnový(1,{neg,comp,masc,pl,acc})...not in D</pre> nejnový(1,{comp,masc,pl,acc}). not in D nejnovější(2,{masc,pl,acc}) =< ...not in D</pre> ... not in D (2 alter.) jnovější(1, {neg,masc,pl,acc}). not in D nejnovější(1,{masc,pl,acc}). ...not in D nejnovější(3,{masc,pl,nom}) = .....D nový(2,{sup,masc,pl,nom}) =<</pre> ....D nový(1,{sup,masc,pl,nom}), ..... D; SOLUTION ... same as 1st alter., but nom instead of acc ... nejnovější(3,{masc,sg,nom}) =< ....Dt in D nový(2,{sup,masc,sg,nom}) =<</pre> ....Dt in D nový(1,{sup,masc,sg,nom}) ..... D; SOLUTION .. same as 1st alter., but sg,nom instead of pl,acc nejnovějšý(3,{masc,pl,nom}) =< .....not in D nejnovějšý(2,{masc,pl,nom}) =< ...not in D (2 alter.) ...not in D nejnovějšý(1,{masc,pl,nom}). jnovějšý(1,{neg,masc,pl,nom}). not in D Fig. 3 

An example of synthesis: we want to obtain s({sup,masc,pl,acc},podlý) -> (podlý,2) e D; see fig. 2

An example of analysis: we want to obtain a(nejnovější#); see fig. 3.

Comment. Better written rules in CRS would not allow for the 4th alternative in the not allow for the 4th alternative in the first step ("nejnovějšý"), because "š" could not be followed by "ý" in any Czech word form; however, constructing the other unsuccessful alternatives could not be a priori cancelled - only the dictionary can decide, whether e.g. "jnový" is or is not a Czech adjective.

Comment on comment. No change in the rules would be necessary if a separate phonology and/or orthography level is used; then, the "šý" possibility, being orthographically impossible, is excluded there, of course.

## 4. Conclusion

This formalism will be probably sufficient for Czech (no counter-example to this thesis has been discovered so far). For inflected words one or two "levels" (i.e., successive rule applications) will suffice, agglutinative elements (e.g., adjective comparison) will probably need three to five rules.

# References

- EBSAT VII (1982): Morphemic Analysis of Czech Prague 1982
- EBSAT VI (1981): Lexical Input Data for Experiments with Czech, Praha 1981 Koskenniemi, K. (1983): Two-level morphology, Univ. of Helsinki, Dept. of Gen. Linguistics, Publications No.11 Hajič, J., Oliva, K. (1986): Projekt česko-
- ruského strojového překladu, (A Project Automatic Extraction of Significant
- Terms from Texts), EBSAT X
- Kirschner, Z. (1987): Kirschner, Z.: APAC3-2: An English-to-Czech Machine Translation System, EBSAT XIII
- Kay, M. (1987): Non-Concatenative Finite-State Morphology, In: Proceedings of the 3rd European ACL meeting, Copenhagen, Denmark, April 1987

EBSAT = Explizite Beschreibung der Sprache und automatische Textbearbeitung, UK Praha