Linguistic **I**ssues in **L**anguage **T**echnology – **LiLT**
Accepted, December 2008

On dissolving morphological long distance dependencies in Russian verbs

Dirk Saléschus
Annette Hautli

Published by CSLI Publications

LiLT volume 1, issue 3 December 2008

On dissolving morphological long distance dependencies in Russian verbs

Dirk Saléschus, *University of Konstanz,*
Annette Hautli, *University of Konstanz*

Abstract

This paper presents implementational issues of a finite-state approach for two crucial parts of Russian verbal morphology: Aspect formation and deverbal nominalization with nie based on aspect formation. The first process involves morphological blocking in order to avoid overgeneralization and is implemented with the xfst- tools via combining two powerful mechanisms - flag diacritics and rewrite rules. This considerably helps reducing the network size while having only small effects on processing time. Deverbal nominalization with nie has also been considered as involving some form of blocking. However, we show how to reanalyze this morphological process as a simple case of phonological neutralization which fits into a broader theory of the Russian sound system. The analysis and implementation presented here are thus theoretically consistent while maintaining implementational effectiveness.

LiLT Volume 1, Issue 3, December 2008.
On dissolving morphological long distance dependencies in Russian verbs.
Copyright © 2008, CSLI Publications.

1 Introduction

Long distance dependencies pose an interesting problem for linguistic theories. Especially morphological analyses based on word syntax approaches can encounter difficulties with them. The reason is that in some cases an affix has to have access to the internal morphological structure of the form with which it combines. One solution is the percolation of features from the inner morphemes to the outer morphemes with some process of feature unification. However, the problem of dealing with percolation constraints without resort to stipulated features has lead some linguists to argue in favour of other frameworks such as, e.g., realizational morphology or parallel approaches like optimality theory. In this paper, we present very simple linguistic analyses which in turn find a very simple implementational counterpart.

This paper presents an analysis of a crucial part of Russian verbal morphology and its implementation via finite-state transducers using the *xfst* tools provided by XEROX and documented in Beesley and Karttunen (2003).[1]

Aspect formation and nominalization with suffixal *nie* can be characterised as long distance dependencies because of the interaction between non-adjacent morphemes. However, when one attempts to provide a computationally simple implementation, it becomes clear that these two morphological phenomena can be reanalyzed as purely local phenomena. In fact, although aspect formation in Russian involves an interaction between lexicon and grammar, we present an implementation which is already sketched in Beesley and Karttunen (2003) and needs only one module, namely a standard finite-state morphology. Our approach neither complicates the grammar architecture nor enlarges the network out of proportion.

The structure of the paper is as follows. Section 1.2. takes a close look at the aspect formation in Russian verbs and gives an introduction to the complex interaction between morphological pre- and suffixation. We describe in what sense these morphological processes involve an interaction between lexical and grammatical meaning and how they are constrained by that interaction. Section 1.3. explores a simple but so far not widely used algorithm within *xfst* for implementing the analysis which makes use of derived flag diacritics. This approach is illustrated with some examples. In addition, the consequences for the network size are evaluated and presented. The discussion of aspect formation pro-

[1] Parts of the present work has been presented at the Finite-State Methods and Natural Language Processing 2007, the Sixth International Workshop, at the University of Potsdam.

vides the basis for section 1.4. where we discuss an analysis by Sadler et al. (1996) of the deverbal nominalization with *nie*, a process that is sensitive to the aspectual marking of the verb. We will explain why the deverbal nominalization serves as a touchstone for competing linguistic theories and how our solution fits into that discussion. Deverbal nominalization with *nie* has been analyzed as involving morphological blocking, whereas we propose an analysis based on phonological neutralization.

2 The Aspectual Category in Russian Verbs

The following discussion will be easier to understand if one keeps in mind that aspect formation in Russian involves three different characteristics of a verb: its lexical meaning, its grammatical meaning and its morphological exponence.

The aspect of Russian verbs, like tense and person, is a grammatical category because every lexical meaning of a verb is obligatorily associated with a system of opposing aspectual meanings (Lehmann, 1999). That means that every verb form of the paradigm is classified as carrying some aspectual meaning. However, the aspectual category in Russian is by no means a prototypical inflectional category. This becomes clear if one applies a number of tests to distinguish grammatical from derivational categories, as presented e.g. in Plank (1991). Additionally, these characteristics most often differ among subgroups of verbs and among subparadigms. Some of these deviations pose interesting problems for analysis and implementation as we will point out in the following.

The aspectual category opposes perfective and imperfective aspect. The majority of Russian verbs morphologically realizes this grammatical distinction, although the morphological exponence is quite complex and allows for very few cross-the-board generalizations. This is typical for derivational categories. Prefixation, suffixation, suppletion, stem allomorphy, and a combination thereof are used. The characterisation of the abstract grammatical meaning varies from author to author but can roughly be characterised as an opposition between completed (perfective aspect) and uncompleted (imperfective aspect) events, with each aspect combining a number of concrete meanings like progressive or iterative meaning, dependent on the context. In the structuralist tradition, the perfective aspect is considered to be the marked aspect because it expresses fewer concrete meanings than the imperfective aspect.

Here we won't consider the particular and intricate semantics of each aspect, see Lehmann (1999), but concentrate on the morphological peculiarities of prefixation and suffixation and the involved interaction of lexical and grammatical meaning. We will show what effects each morphological change of a verb has. The following overview of the morphological exponence can be found in any grammar of Russian, e.g., Timberlake (2004).

Let us start with simple verb stems. The overwhelming majority of them are imperfective, however, there also exist perfective simple stems. The aspect of simple stems is thus an idiosyncratic property and has to be marked for each single stem.

2.1 Lexicon vs. Grammar: Prefixation and Suffixation

Instead of describing the morphological exponence of the aspectual opposition for both kinds of simple verb stems we want to describe more generally two morphological processes that can be applied to them: prefixation and suffixation. Lexical prefixation of verbs is quite productive in Russian. There exist two kinds of prefixation, lexical and grammatical. Consider first the formation of complex verb stems by means of lexical prefixation.

It is a derivational process and leads to the formation of both new lexemes (lexical relation) and complex stems (morphological relation). From a semantic point of view this process of lexeme formation can either be opaque or can lead to transparent composed meanings as, e.g., in the case of different semantic classes called *Aktionsarten.* In the following examples[2] semantically transparent prefixation is contrasted with opaque lexical prefixation[3]:

(1)	*nosít'*	(carry-indet.ipf.)	*vnosít'*	(carry in-pf.)
	vynosít'	(carry out-pf.)	***iznosít***	**(to wear out-pf.)**
(2)	*begát'*	(run-indet.ipf.)	*vbegát*	(run inside-ipf.)
	vybegát	(run out-ipf.)	***izbegát***	**(avoid-ipf.)**

[2] The following transcription conventions are adopted here: the *y* stands for the high back unrounded dorsal [ɨ]. A soft consonant is a consonant that is palatal or has secondary palatalization. The latter feature is signalled by an apostrophe after the consonant (e.g. *t'*). The softness of consonants is predictable when they are followed by the front vowels *i* or *e* and is left out in these contexts. The symbols č, š and ž stand for a soft alveo-palatal affricate [tʃ] and the posterior voiced and unvoiced fricatives [ʃ] and [ʒ], respectively. At the surface, the č is always soft whereas š and ž are always hard. Finally, an accent signals stress.

[3] All forms are in the infinitive, unless indicated explictly. Simple verbs of motion have an additional category of (in)determinacy, as defined in Tim04.

The items in bold are lexicalized. The meaning of the other items is composed of the meaning of the prefix and the stem. The following example for semantically transparent lexical prefixation with the ingressive *Aktionsart* is from Isačenko (1995:388 f.):

(3)	*govorít'*	(speak-ipf.)	*zagovorít*	(start speaking-pf.)
	igrát'	(play-ipf.)	*zaigrát*	(start playing-pf.)
	kričát'	(cry-ipf.)	*zakričát*	(start crying-pf.)

There are around 20 prefixes which can be used for lexical prefixation with both perfective and imperfective simple verb stems. Lexical prefixation can also be applied cyclically, leading to complex forms such as:

(4)	*polnít'*	(colloquial:fill-pf.)	*výpolnít'*	(fulfil-pf.)
	perevýpolnit'	(overfulfil-pf.)		

This phenomenon is also found in other languages like English or German which forms the verbs *füllen* (to fill), *erfüllen* (to fulfil), and *übererfüllen* (to overfulfil), respectively.

Note, however, that not every complex stem is formed from an actual existing base stem. There are verbs like *dobávit* (fill-pf.), *pribávit* (add-pf.), *zabávit* (amuse-pf.), without a verb *bávit.* Even though they look like complex stems they have to be analyzed as simple stems. This is similar to English morphology with verbs like *perceive*, *receive* with no existing word *ceive* (see Spencer (1998:85f.) for a discussion of such examples).

Additionally, some stems have only one or a few actual prefixed variants whereas others combine with many prefixes (Isačenko, 1995:357). See, e.g., the possible lexical derivations of the stem *xodít'* (go-indet.ipf.):

(5) *vxodít', vsxodít', vyxodít', doxodít', zaxodít', isxodít', naxodít', obxodít', otxodít', perexodít', poxodít', podxodít', prixodít', proxodít', rasxodít's'a, sxodít', uxodít'*; only exception: **nadxodít'.*

The lexicon that we built is fully productive and contains all actual and potential complex word forms. In the course of this paper we only examine morphological constraints on word forms and put aside semantic restrictions, which should be dealt with in a separate module or lexicon.

From a grammatical point of view there is one important grammatical byproduct associated with lexical prefixation. All newly formed lexemes are always perfective stems. In other words, lexical prefixation always leads to perfectivization[4]. This shows the intricate connection between lexical and aspectual meaning. For a deeper discussion of these mechanisms, we refer the reader to Breu (1994).

2.2 Aspect Formation with Prefixation

Now we focus on the exponence of aspectual opposition, which includes suffixation in addition to grammatical prefixation (ignoring other means of morphological exponence). Normally, Russian verbs realize the aspectual opposition by two different stems (also called partner verbs) — one perfective and one imperfective. Both stems together make up an aspectually complete verbal lexeme. Since the aspectual opposition is not expressed via exponence on the same stem (as is the case for other grammatical categories like number or person) this morphological process is called grammatical derivation (Breu, 2000). In that respect, the Russian aspect deviates again from a prototypical inflectional category like person and number, illustrated in the following example:

(6)	*govorít’*	(to speak-ipf.inf.)
	govor’ú	(to speak-ipf.pres.1.sg.)
	govoríš	(to speak-ipf.pres.2.sg.)
	govorít	(to speak-ipf.pres.3.sg.)

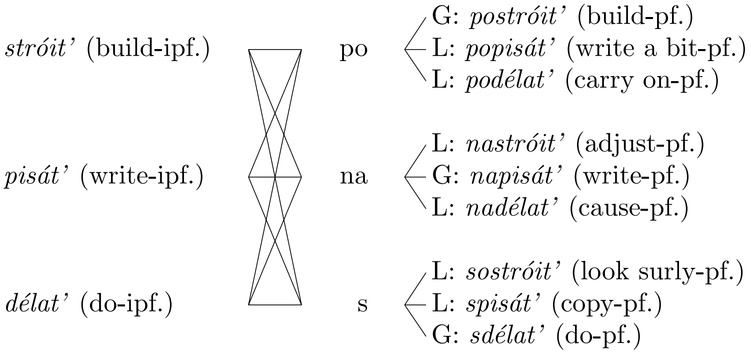
Simple imperfective verb stems freely combine with the same set of prefixes as simple perfective stems. However, there is one important difference: imperfective stems use prefixation not only to build new lexemes but also to express the perfective partner verb and thus build an aspectually complete lexeme. In that case one talks about grammatical prefixation, as opposed to lexical prefixation[5].

From a semantic point of view, grammatical prefixation never alters the lexical meaning. From a morphological point of view, a new complex stem is formed. The crucial fact is that for each simple im-

[4] There are just a few regular exceptions, most often with verbs and prefixes which were borrowed some centuries ago, as, e.g., with the prefix *bez*, meaning ‘without’.

[5] Some authors, as e.g. Breu (1994), have argued that every prefixation leads to a change in the lexical meaning. However, there are aspectually relevant contexts where the morphological opposition between imperfective and perfective aspect is neutralized but where the lexical meaning stays the same. This is the case in, e.g., the historical present which allows only imperfective forms. In these contexts prefixed perfective verbs can have simple imperfective stems as their counterpart. This would not be possible if prefixation had changed the lexical meaning. Breu concluded that in these cases, the prefix is lexically redundant for the verb lexeme and consequently it is semantically empty. Note that the neutralization of the aspectual opposition in certain morphosyntactic constructions is a characteristic of an inflectional, not of a derivational, category.

perfective stem there is exactly one prefix which is used exclusively for grammatical prefixation. All remaining prefixes are used for lexical prefixation only. The choice of the grammatical prefix that can combine with a given imperfective simple stem is not predictable and has to be marked for every simple imperfective stem. The following sketch with some prefixes and stems illustrates this. The “G” stands for grammatical prefixation, “L” stands for lexical prefixation. One starts out with a simple stem (seen on the left hand side), which can then combine with any of the listed prefixes and build a complex stem (right hand side).


stróit’ (build-ipf.)
pisát’ (write-ipf.)
délat’ (do-ipf.)
po
na
s
G: *postróit’* (build-pf.)
L: *popisát’* (write a bit-pf.)
L: *podélat’* (carry on-pf.)
L: *nastróit’* (adjust-pf.)
G: *napisát’* (write-pf.)
L: *nadélat’* (cause-pf.)
L: *sostróit’* (look surly-pf.)
L: *spisát’* (copy-pf.)
G: *sdélat’* (do-pf.)

2.3 Aspect Formation with Suffixation

In traditional analyses, simple perfective verb stems can change the stem vowel to express imperfective partner stems. We analyze this phenomenon as a case of suffixation. There are two allmorphs of the imperfective suffix: an empty V-slot and the string *yv*. If not filled by an adjacent vowel from some suffix, the empty slot is per default filled with the vowel *a*. The correct allomorph of the imperfective suffix is determined by morphological class membership. One generalization is that the morpheme which combines with simple perfective stems to express the imperfective aspect is always the same, namely *a*:

(7)	throw:	*brósit’*	(pf.)	*brosát’*	(ipf.)
	deprive:	*lišít’*	(pf.)	*lišát’*	(ipf.)

We analyze the *i* in *brosít’* as an aspectual suffix, because it seems to be in complementary distribution to the imperfective suffix *a*. However, we analyze it as a thematic vowel for one morphological class of simple perfective stems. The root is then *bros* and *liš*, respectively. The motivation for our morphological analysis is that in Russian morphological verb classes assign thematic vowels to stems of a paradigm in different ways. Sometimes this vowel is kept in only some stems of the paradigm, sometimes in almost all stems. This separation of thematic vowels from roots facilitates the analyses of the imperfective suffix and of the deverbal nominalization.

2.4 Aspect Formation - Secondary Imperfectivization

Complex perfective verb stems also use suffixation. For these stems this process is called secondary imperfectivization. This is a grammatical process only and is never possible for simple imperfective verb stems. Complex perfective stems normally take the allomorph *yv* but some have the allomorph *a* and some even can have both (so called aspectual triples)[6]:

(8)	manufacture:	*izgotóvit'*	(pf.)		
		izgotovl'át'	(ipf.) or	*izgotávlivat'*	(ipf.)

Finally, some (simple or complex) perfective stems show consonant alternations when imperfectivized while others do not. This also has to be marked lexically:

(9)	stem allomorph:	render:	*javít'* (pf.)
			javl'át' (ipf.)
		manufacture:	*izgotóvit'* (pf.)
			izgotávlivat' (ipf.)
	no stem allomorph:	throw:	*brósit'* (pf.)
			brosát' (ipf.)
		copy:	*perepisát'* (pf.)
			perepísyvat' (ipf.)

The following algorithm summarizes the possible morphological processes, their conditions and effects with some examples. Note the identity of the different lexemes, here abbreviated as Lex 1 and Lex 2.

[6] The /l/ in some of the verb stems is just part of the stem allomorph and bears no extra meaning. It is the result of a palatalization process in Proto-Slavonic which lead to /l/-epenthesis after palatalized labial consonants.

Derivation cycle for Russian simple verbs

```
If root is perfective (Lex 1)            (Ex.: bros-i-t' – throw-pf.)
    If prefixation
        Then: result is a new perfective lexeme (Lex 2)
            vy-bros-i-t'                 (throw out-pf.)
        Then: imperfectivization via secondary suffixation (Lex 2)
            vy-bras-yv-a-t'              (throw out-ipf.)
    Else: imperfectivization via suffixation (Lex 1)
            bros-a-t'                    (throw-ipf.)

Elsif root is imperfective (Lex 1)       (Ex.: pis-a-t' – write-ipf.)
    If prefixation is lexical
        Then: result is a new perfective lexeme (Lex 2)
            s-pis-a-t'                   (copy-pf.)
        Then: imperfectivization via secondary suffixation (Lex 2)
            s-pis-yva-t'                 (copy-ipf.)
    Else: prefixation is grammatical (Lex 1)
        na-pis-a-t'                      (write-pf.)
                                    not allowed:
        imperfectivization via secondary suffixation (Lex 1)
        *na-pis-yv-a-t'                  (write-ipf.)
```

3 Blocking affixation with derived flag diacritics

The crucial question is how (secondary) imperfectivization can be blocked for grammatically prefixed perfective complex stems (see last line in the algorithm above). We will first sketch the solution informally and then consider possible implementations.

The imperfective suffix needs two kinds of information. First, is the stem perfective or imperfective? Suffixation is only possible for perfective verb stems. The second question is whether the verb is lexically prefixed. Secondary imperfectivization applies only if the complex stem is created by lexical prefixation, not by grammatical prefixation. Thus, imperfectivization is accomplished in order to not just create imperfective verb stems but to create imperfective partner verbs, i.e. aspectually complete lexemes.

How can this be captured in a morphological framework? How can a morphological type of affixation be made sensitive to the complex morphological as well as lexical structure of a verb stem?

One possibility is the following. We assume that the imperfective suffix only combines with a stem which is not yet aspectually complete, i.e., where the perfective stem is missing an imperfective counterpart in the paradigm. We assume further that every stem can signal whether it is aspectually complete.

Next let us assume that every simple imperfective stem is marked for its matching grammatical prefix (if there is one at all). Let us call this marking the stem-prefix-feature. Once this prefix is encountered, the lexeme is saturated and signals that it is aspectually completed by setting the stem-prefix-feature. The imperfective suffix is then blocked from application by reference to that feature. Thus, if the prefix and the stem-prefix-feature match, then a new feature is set and suffixation is blocked by that new feature.

3.1 Implementational Approaches

We first provide a very brief introduction to Finite-State Transducers (FST), we then present three possible FST implementations of the Russian data discussed above (sections 1.3.1 and 1.3.2), and we then compare the approaches (section 1.3.3).

Finite-state transducers are one of the main concepts in computational phonology and morphology (Jurafsky and Martin, 2008). They are simple machines, which consist of states connected by arcs. Each arc is labeled by a pair of symbols, an input and an output symbol. When applied to morphology, each word can be modeled by such a transducer, which sets the underlying form and morphological information in relation to the surface form of a word. Such a finite-state transducer is bidirectional, hence it can analyze word forms but also generate them. For a more technical introduction we refer the reader to Roche and Schabes (1997).

When using FSTs for computational morphology, there are several ways to implement the data discussed above. Beesley (1998) discusses different strategies, among them using concurrent rule transducers as in two-level morphology or composing in constraints at compile time. The first solution has the disadvantage of slower performane at runtime whereas the latter solution leads to an enormous increase in network size.

Beesley favours a solution with flag diacritics (also described in Beesley and Karttunen (2003)). Flag diacritics are formulated in a template, which comprises three parts: operation, attribute, and value. The operations include P (positive set), N (negative set), R (require), U (unify) and D (disallow). The first two always succeed in setting the value of an attribute. The last three will fail if the attribute does not have the right value.

Flag diacritics are part of the normal alphabet insofar as they are interpreted as epsilons and can be added to lexical entries and state constraints on the concatenation of strings. However, the enhanced *xfst* lookup routines process them in a special way and enforce the dependencies between morphemes. The lookup routines do this by introducing a small amount of memory which suffices to capture the long distance dependencies (Beesley, 1998:123). With flags, it is in principle possible to create a blatantly overgenerating lexicon and let the lookup routines rule out impossible or undesired combinations. The only disadvantage of flags is a possible slower performance due to backtracking.

We try to solve the problem of the Russian data by reference to flag diacritics. The solution seems quite obvious: The flag of the imperfective suffix interacts with the flags from stem and prefix according to well-defined conditions. This is broken down into two steps: First the flags of the prefix and the stem interact. The result is then handed over to the flag of the imperfective suffix. There are several kinds of flags triggering different processes and so again there are several possible strategies for an implementation of flags. We will not describe the whole flags inventory in *xfst*; for detailed information we refer the reader to Beesley and Karttunen (2003).

We illustrate our implementation with the example *pisat'* (write-ipf.).

(10)	*pisat'*	(write-ipf.)	*napisat'*	(write-pf.)
	*napisyvat'	(write-ipf.)		

The simple imperfective verb stem *pisat'* takes on a grammatical prefix in order to express the perfective form *napisat'*. Due to grammatical prefixation, secondary imperfectivization is ruled out _*napisyvat'_. Contrast this to lexical prefixation, where secondary imperfectivization is obligatory in order to express imperfective meaning.

(11)	*pisat'*	(write-ipf.)	*spisat'*	(copy-pf.)
	spisyvat'	(copy-ipf.)		

Working exclusively with flag diacritics, we assign an individual flag to every prefix signaling a positive value, e.g. `@P.NA.PLUS@` for the prefix *na* and similarly `@P.U.PLUS@` for the prefix *u*. The first `P` stands for an operation over the feature, in this case positive setting of the feature. `NA` is the name of the feature and `PLUS` is its value. The whole expression is surrounded by @-signs.

Stems are also assigned flags. A stem flag checks the value of the prefix flag and resets it only if the prefix combines with this stem for grammatical prefixation. For example, the grammatical prefix of the stem *pis* is *na* and therefore changes its flag value. This is achieved by the flag `@N.NA.PLUS@`. Here `N` signals negative resetting of the value ot the `NA` feature such that the value is reset to the complement of `PLUS`. The value of another prefix like *u* is left intact by the stem *pis*.

The imperfective suffix, finally, is also assigned a flag with a value set to `PLUS`. It has to list all possibilities like `@R.NA.PLUS@`, `@R.S.PLUS@` etc.

A simplified example illustrates how this works.

The following expressions illustrate possible concatenations in *lexc*[7] (concatenation is accomplished via continuation classes (linked sublexicons). They are here marked with a plus for clarification):

```
na@P.NA.PLUS@+pis@N.NA.MINUS@+[yv@R.NA.PLUS@ |
@R.S.PLUS@...]

s@P.S.PLUS@+pis@N.NA.MINUS@+[yv@R.NA.PLUS@ |
@R.S.PLUS@...]
```

In the first case the value of the first flag is set to `PLUS`. This value is reset by the stem flag to `MINUS`. As a result, the flag of the imperfective suffix requiring a `PLUS` value does no longer match with this complex stem. In the second case the value of the first flag is again set to `PLUS`. This time it is left intact by the stem flag and the imperfective suffix can successfully check for a `PLUS` value. This gives us exactly the right results.

With the sketched solution, the imperfective suffix would have flags attached with it which check for every possible prefix. This is expressed by the disjunctive listing of flags for the imperfective suffix, indicated by "|". One could also have multiple entries for the imperfective suffix, each one bearing only one flag. However, since flag diacritics lead to a runtime penalty, we offer a modified and more elegant solution with a reduced number of flags, which is described in the following section.

[7] *lexc*, for *Lex*icon *C*ompiler, is a high-level declarative language and associated compiler for defining finite-state automata and transducers; as its name indicates, it is particularly well suited for defining natural-language. It is provided by the **XEROX** finite-state tools (Beesley and Karttunen, 2003). The expressions surrounded by plus-signs are morphemes and the plus-signs indicate their linear concatenation.

3.2 Deriving Flag Diacritics

There is a second strategy which uses a combination of flags and continuation classes by doubling the entries for stems. To take the example from above, the prefix *na* takes again the flag `@P.NA.PLUS@`. The first entry for the stem *pis* has the flag `@D.NA.PLUS@` where the `D` indicates that the feature `NA` is not allowed to have the value `PLUS`. Thus the stem *pis* may combine with any prefix except the one it uses for grammatical prefixation, namely *na*. The continuation class of that stem is the imperfective suffix. The second entry for *pis* has the flag `@R.NA.PLUS@`. Here the `R` indicates the requirement for a preceding flag with the feature `NA` set to the value `PLUS`. The absence of a preceding flag or any other flag setting is forbidden. The continuation class of that stem entry can be anything except the imperfective suffix. The obvious disadvantage of that solution is the increase in network size by doubling information.

There is a third solution with single entries for all morphemes and a minimal number of flags used that is already sketched in Beesley and Karttunen (2003:351). In this analysis, all morphemes in the overgenerating *lexc* grammar have a special formal marking. Rewrite rules check the markings of the morphemes and change them into flags in special contexts. To take a concrete example, the prefix *na* has the notation `naPLEX`, the prefix *u* has the form `uPLEX`, and the prefix *s* has the form `sPLEX`. `PLEX` is just a placeholder for any lexical prefix without special meaning.

Stems also have some special formal marking which indicates the prefix that is used for grammatical prefixation. For example, the stem *pis* has the form `pisNA`, indicating that the prefix *na* is used for grammatical prefixation. Similarly, the stem *sluš* is notated as `slushU`. Finally, the imperfective suffix also has an additional formal marking, namely `IMPRLEX`. All these additional formal markings are just menmonic placeholders and could be transformed into a more sophisticated notation according to the linguists needs. When the *lexc* grammar is compiled it contains all conceivable combinations of prefixes, stems and suffixes, among them unwanted combinations. In a next step simple rewrite rules delete the special formal marking of the prefix if the prefix happens to have the same form as the extra formal marking of the stem. Rewrite rules are given here in the *xfst* formalism. They denote regular relations which can later be composed with the lexicon.

To take the above mentioned example with grammatical prefixation, *pisat'* and *napisat'* (to write), one can see how we specify the underlying form (the spaces are not part of the underlying form, they only mark morpheme boundaries):

```
naPLEX pisNA IMPRLEX atj
```

The form `naPLEX` is replaced by simple `na` if somewhere after the prefix the form `NA` is found:[8]

```
define rule1 [n a P L E X -> "na" || _ $[N A] ];
```

This results in the following form:

```
na pisNA IMPRLEX atj
```

A later replace rule transforms all special formal markings which have not been deleted into flag diacritics.

```
define rule2 [R L E X -> "@R.LEX.LEX@"] ;
```

The new intermediate form now looks like:

```
na pisNA IMP@R.LEX.LEX@ atj
```

Two more clean-up rules instantiate the final morphological form.

```
define rule3 [I M P ->"yv"] ;
define rule4 [N A ->"0"] ;
```

The final surface form is:

```
na pis yv@R.LEX.LEX@ atj
```

This form is filtered out by the lookup routines, because the requirement of the flag is not fulfilled. This is exactly what we intended to do. The imperfective suffix cannot combine with grammatically prefixed stems because in the complex stems the prefixes do not have the required flag. The long distance dependency between prefix and suffix is thus resolved into a local interaction between two flags.

In contrast to the example for morphological blocking, we present an example for lexical prefixation and secondary imperfectivization with

[8] In the rules below we follow the writing conventions of Beesley and Karttunen (2003) insofar, as the single symbols of strings are separated by whitespace.

spisat' (to copy) in the following steps.

```
sPLEX pisNA IMPRLEX atj
```

The form sPLEX is not replaced by simple s because the following rule does not apply:

```
define rule1 [s P L E X -> "s" || _ $[S ];
```

However, a later replace rule transforms all special formal markings which have not been deleted into flag diacritics.

```
define rule2 [R L E X -> "@R.LEX.LEX@"] ; define rule3 [P
L E X -> "@P.LEX.LEX@"] ;
```

The new intermediate form now looks like:

```
s@P.LEX.LEX@ pisNA IMP@R.LEX.LEX@ atj
```

Two more clean-up rules instantiate the final morphological form.

```
define rule4 [I M P ->"yv"] ;
define rule5 [S ->"0"] ;
```

The final surface form is:

```
s@P.LEX.LEX@ pis yv@R.LEX.LEX@ atj
```

This example shows that with lexical prefixation, secondary imperfectivization is not ruled out due to the matching flags.

3.3 Consequences for the Network

What are the advantages of this solution? The following graphs (figures 1 to 4) show the network properties of the three implementational strategies.

Our question was the following: How would the different implementational strategies affect network size and runtime performance? To test this we started with a toy lexicon of 3 stems and added one stem after another until we reached a size of 20 stems. Each time we compared the network parameters states (figure 1), arcs (figure 2) and Kb (figure 3). The number of prefixes was held constant at a number of three. The runtime performance of all networks was tested on a Mac OS X, 1.42 GHz PowerPC G4, 768 MB DDR SDRAM.

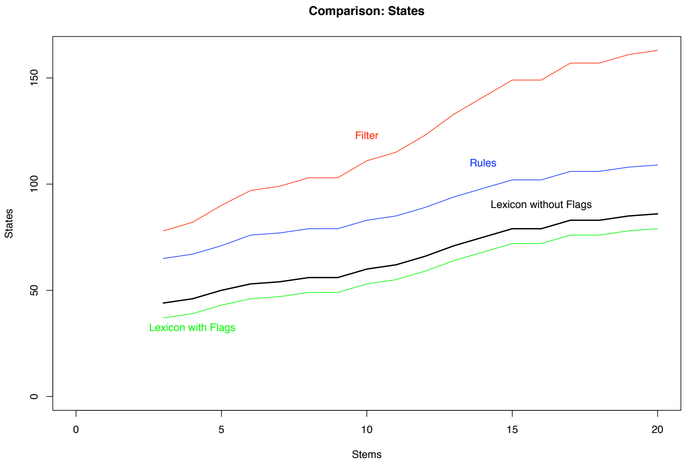


FIGURE 1 Results of different analysis in States

The first thing to note when looking at the network size in states, arcs and Kb (figures 1 to 3, respectively) is that while increasing lexicon the exclusive use of flags leads to the smallest network. This is followed by the strategy with derived flags and finally by the strategy using filters. Interestingly, while increasing the lexicon the increase in network size is the same for the first two strategies but is bigger for the filter. Thus, the difference in network size between the first two strategies will become less and less important when increasing the lexicon to a more realistic size (this could be around 3,000 stems). Regarding only network size, it seems clear that the use of filters is the less optimal solution.

But why does the strategy with flags lead to the smallest network? There is a simple explanation. In *xfst*, lexicons are normally developed with the *lexc* tool. This is a high-level language especially for specifying lexicons (Beesley and Karttunen, 2003:203) which are compiled into simple transducers. Any linguistic rules, be they phonological or orthographic, are specified in other files and the final morphological transducer of a language is then created by composing in our case two transducers into one network which is normally bigger in size then any of the single transducers.

Since flags are normally defined in the lexical transducer, no further composition with a rule transducer is needed. With other implementa-

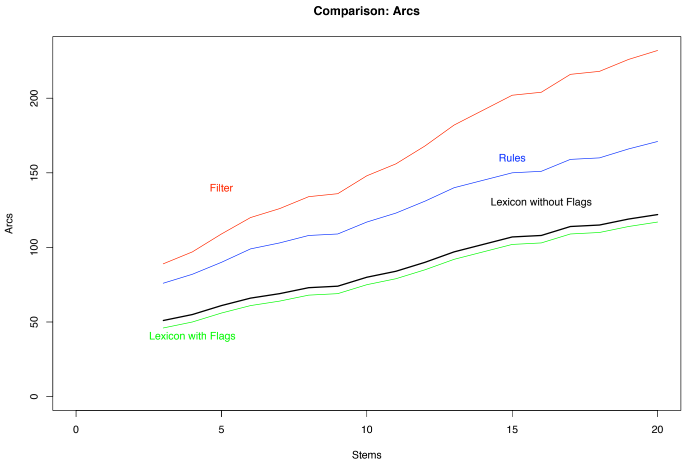


FIGURE 2 Results of different analysis in Arcs

tional strategies, however, first a lexical transducer is specified and then filters or rules to derive flags are compiled into a transducer. Finally both transducers are composed into a new transducer. This explains why using exclusively flags results in a smaller network size. To better assess the influence of flags on the network size one only needs to compare the two lexical transducers that we used for our implementations. The lexical transducer used for filters and rules is smaller than the lexical transducer incorporating flags.

Still, the question remains whether deriving flags could have an advantage over putting flags directly into the lexical transducer. The point is that we have dealt with only a small part of Russian morphology and trying to come up with a bigger picture would sooner or later require the use of a rule transducer. Once the lexical transducer with build-in flags is composed, the network size will become at least equal to the approach with derived flags.

With that discussion in mind, it becomes clear why it is also difficult to evaluate the consequences for runtime performance with networks of different size (see figure 4).[9] To test this we used the following methodology.

First we created 10 testsuites of increasing size, ranging from 100,000 randomly generated wordforms up to 1,000,000. With the command

[9] We owe special thanks to Lauri Karttunen who gave us valuable hints regarding the comparison of runtime performances.

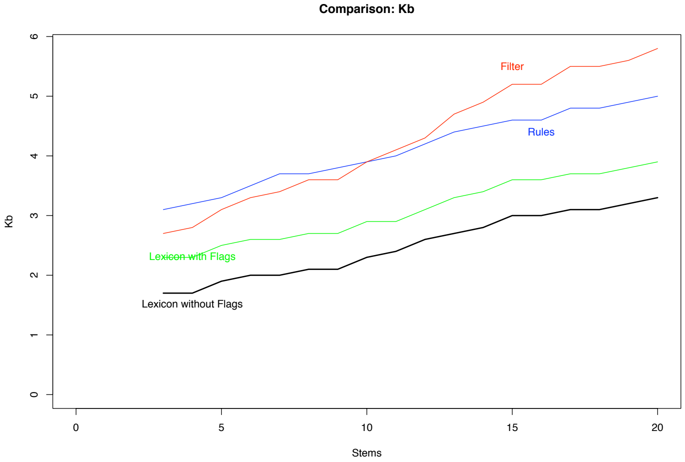


FIGURE 3 Results of different analysis in Kb

"set use-timer on", we could be given the execution time of each operation. We then analyzed each testsuite 10 times to account for minor deviations in the execution time and calculated the average. Finally, we took the average times of the 10 testsuites and calculated the average over all testsuites again.

In the first session the lexical transducers contained 5 stems. In the following sessions we increased the number of stems by 5 till we reached 30 stems. These numbers can be seen in figure 2. As expected, the implementation using filters performed better than the solution with rules. The better performance of flags can be explained, as mentioned above, by the fact that there is only a single transducer instead of a composed network of two transducers. The difference in average runtime performance for all three implementations stays constant at around 1.5 sec with increasing lexicon size. Therefore, one can say that the three networks lie within one range of performance. The marginal difference can be neglected when building larger lexicons and does not play a major role in the evaluation.

3.4 Interim Summary

Our strong preference lies with derived flags diacritics. First of all, it keeps the network size small and therefore compilation time is reduced. At the same time, runtime performance is not negatively effected. Second, it provides an elegant solution to the problem, a solution that

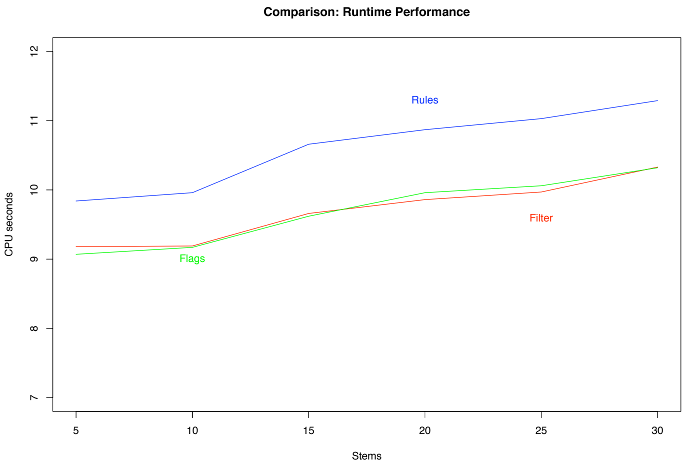


FIGURE 4 Results of different analysis in CPU seconds

captures a linguistic generalization. Third, it could easily be embedded into a large-scale grammar of Russian morphology.

So far we have shown how to block a special case of suffixation. The long distance dependency was resolved into two local phenomena. First, the concatenation of prefixes and stems was checked for some kind of pattern matching. Second, the concatenation of the complex stem and the imperfective suffix was restrained by flag diacritics. In the next section a long distance dependency is again resolved into a purely local phenomenon.

4 Russian deverbal nominalization with /nie/

Sadler et al. (1996) describe a special case of blocking in Russian. Almost all simple or complex Russian verb stems as described above can combine with the deverbal nominalization suffix *nie*. See the following examples:

	Verb	**Nominal**	**Nominalization Type**
(12)	*pisát'* (write-ipf.)	*pisánie* (writing-N.)	RES/CEN
(13)	*pét'* (sing-ipf.)	*pénie* (singing-N.)	SE/CEN
(14)	*sobrát'* (collect-pf.)	*sobránie* (meeting-N.)	RES/SE
(15)	*starát's'a* (try-ipf.)	*staránie* (endeavour-N.)	SE
(16)	*raspisát'* (write out-pf.)	*raspisánie* (timetable-N.)	RES
(17)	*spisát'* (write off-pf.)	*spisánie* (writing off-N.)	CEN
(18)	*zatverdét'* (harden-pf.)	*zatverdénie* (hardening-N.)	RES/CEN
(19)	*izgotóvit'* (manufacture-pf.)	*izgotovlénie* (manufacture-N.)	SE/CEN

The type of the resulting nominalization (adopted from Sadler et al. (1996) and given at the end of the line) is not predictable and can be a complex event nominal (CEN, e.g. *spisánie*), a simple event nominal (SE, e.g. *staránie*), or a result nominal (RES, e.g. *raspisánie*). One nominalization can also have different types. We will not provide more details about the semantic characteristics and the tests for different types of nominalizations because they are not crucial for the following discussion, for further reference see e.g. Grimshaw (1990).

Sadler et al. (1996:193) mention two generalizations that apply to this kind of word formation. First, if nominalization applies at the secondary imperfectivization of a lexically prefixed verb then the type of the nominalization is always the same, a complex event nominal.

Second, if the secondary imperfectivization of a lexically prefixed word does not use the allomorph *yv* but instead the allomorph *a*, then the deverbal nominalization suffix *nie* cannot be attached to that verb. As examples for that blocking, they cite the following data from lexically prefixed verbs:

(20)	proclaim:	*provozglasít’* (pf.)	*provozglašát’* (ipf.)
	proclamation:	*provozglašénie*	**provozglašánie*
(21)	visit:	*posetít’* (pf.)	*poseščát’* (ipf.)
	visit (N):	*poseščénie*	**poseščánie*
(22)	inform:	*soobščít’* (pf.)	*soobščát’* (ipf.)
	communication:	*soobščénie*	**soobščánie*
(23)	consolidate:	*ukrepít’* (pf.)	*ukrepl’át’* (ipf.)
	consolidation:	*ukreplénie*	**ukrepl’ánie*
(24)	destroy:	*razrušít’* (pf.)	*razrušát’* (ipf.)
	destruction:	*razrušénie*	**razrušánie*
(25)	destroy:	*razorít’* (pf.)	*razor’át’* (ipf.)
	destruction:	*razorénie*	**razor’ánie*
(26)	resolve:	*postávit’* (pf.)	*postavl’át’* (ipf.)
	resolution:	*postanovlénie*	**postanovl’ánie*

This is ostensibly a case of long distance dependency. The suffix *nie* has to have access to the morphological structure of the morphologically complex verb.

Sadler et al. (1996:203) discuss this problem from the point of view of word syntax. According to their analysis such an approach in combination with locality conditions on affixation has to abuse feature marking and percolation conventions to “permit a purely morphological feature to percolate from the root to the top of the tree”. Even then it does not explain blocking effects of deverbal nominalizations in Russian verbal morphology. It is argued that the generalization can only be stated by a morphological rule of referral (for further reference, see Stump (1993)).

Karttunen (2003) has already shown that rules of referral are no more powerful than regular relations. However, we would like to suggest another and much simpler analysis of the Russian data.

4.1 Reanalysis of blocking

The crucial assumption by Sadler et al. (1996:192) is: There is no purely phonological restriction which will account for the lack of **razrušánie*, **ukrepl’ánie* [...].

But let us take a closer look at the data. The first thing to note is that if a verb contains stems with palatalized allomorphs in its paradigm then the deverbal suffix *nie* is always attached to such a stem allomorph:

(27)	*izgotóvit'*	(manufacture-inf.pf.)
	izgotóvl'u	(manufacture-1.sg.pres.pf.)
	izgotovlénie	(manufacture-N.)

(28)	*provozglasít'*	(proclaim-inf.pf.)
	provozglašú	(proclaim-1.sg.pres.pf.)
	provozglašénie	(proclamation-N.)

The next thing to note is that the suffix *nie* is always added to a stem ending either in *a* or *e*, even though the stem to which the deverbal nominalization is added, might never realize that vowel with other stems elsewhere in the paradigm:

(29)	*izgotóvit'*	*izgotovlénie, *izgotovlínie*
	ukrepít'	*ukreplénie, *ukreplínie*

A simple assumption leading to an elegant and intuitive solution is that the vowel preceding *nie* in fact belongs to the nominalizing suffix. It is attached to a stem that does not have a thematic vowel and is realized as *a* after hard consonants and as *e* after soft consonants. The only morphological requirement for the application of the nominalization suffix is to take a special palatalized stem form if there is one in the paradigm. Everything else is governed by phonology (in the rules below, V stands for vowel):

(30)	/Vnie/	→	[enie]	/ C[soft] _	
	/Vnie/	→	[anie]		elsewhere

This analysis can also be stated in a more sophisticated manner. Using underspecified feature structures, one possibility would be to say that the underlying vowel needs only to be specified for [LOW]. A postlexical rule deletes this feature in the context of a preceding soft consonant which is always specified for [HIGH] (Lahiri and Evers, 1991). A vowel with no feature specification at all will per default be realized as coronal [e]. With [LOW] after hard consonants as the only specification, this vowel will be realized as dorsal [a] by a redundancy rule.

There is one slight complication with sibilants in Russian. In Russian, all consonants can have soft (with secondary palatalization) and hard variants (without secondary palatalization). The sibilants *š*, *ž* and the dental affricate *ts*, however, do not have both variants but always surface as hard consonants. However, in certain phonological contexts which are sensitive to the softness of the consonant, the hard sibilants behave like soft consonants. This is true for the above mentioned rules of vowel alternation and for similar rules of stress-sensitive vowel neutralization. There is thus evidence that underlyingly the sibilants are soft. Thus, the examples with sibilants do not present counterexamples to our analysis, because they have to be seen in the light of a complete phonological analysis of Russian.

With that assumption one can explain why the form *provozglašénie* is encountered instead of *provozglašánie*. Again, only a detailed phonological analysis will lead to these generalizations. The exact details of the Russian phonological system are quite elegant and straightforward but need not to be copied one by one into the *xfst*-framework. It suffices to know that the generalization to be captured is phonological. With these observations at hand one can now explain the following blocking effect:

(31)	*razrušít'*	(destroy-inf.pf.)	*razrušénie*	(destruction)
	razrušát'	(destroy-inf.ipf.)	**razrušánie*	

(32)	*razorít'*	(destroy-inf.pf.)	*razorénie*	(destruction)
	razor'át'	(destroy-inf.ipf.)	**razor'ánie*	

The reason why a form like **razoránie* is never encountered as opposed to the form *razorénie* is because there is a simple case of phonological neutralization at work. The underlying stem used for the formation of razorenie is /razor'/ which ends in a soft consonant. The vowel of the derverbal suffix undergoes a simple assimilation after soft consonants it is fronted and surfaces as [e] whereas elsewhere it surfaces as [a].

This phonological generalization also holds for verbs from the consonantal class which do not have imperfective suffixes but signal the secondary imperfectivization via the whole stem form as shown in the following two examples:

(33)	*sobrát'*	(collect-inf.pf.)
	sobránie	(collection)
	sobirát'	(collect-inf.ipf.)
	sobiránie	(collecting-N)

(34)	*výžat'*	(wring out-inf.pf.)
	**vyžánie*	not attested
	výžimat'	(wring out-inf.ipf.)
	vyžimánie	(wringing out-N)

The phonological generalization also applies to unprefixed verbs:

(35)	*pisát'*	(write-ipf.)	*pisánie*	(writing-N)
	bít'	(beat-ipf.)	*bijénie*	(beat-N)

The last piece of evidence comes from semantics. Normally, nominalizations with *nie* formed from perfective verbs do not show a predictable pattern of nominalization type, as Sadler et al. (1996:190) point out. It is therefore interesting to note that nominalizations of lexically prefixed verbs where the secondary imperfectivization uses the allomorph *a* almost always have a complex event reading besides a simple event reading or result reading. The explanation is easy in our analysis: the nominalization of these verbs have two potential underlying stems (perfective and imperfective). Given that the secondary imperfectives show a regular pattern of nominalization type this generalization is preserved independent of the phonological neutralization.

4.2 Implementational issues of nominalization

A reformulation of the phonological rewrite rules above in *xfst* looks as follows:

```
define Csoft [ p' | t' | k' | ... ] ;
define rule1 [ V n i e -> e n i e || Csoft _ ] ;
define rule2 [ V n i e -> a n i e ] ;
```

One of the advantages is that these rules in combination with the underspecified representation can easily be added to a large-scale grammar.

To sum up, with a different morphological segmentation, the effect of morphological blocking turns into a case of local phonological neutralization.

5 Conclusion

We started out with the aim of implementing a basic part of Russian verbal morphology with complex interactions between lexicon and grammar. In doing so, it was possible to reanalyze the data and show how computational and theoretical issues can positively influence each other in such a way that computationally simple solutions correspond to concise linguistic generalizations. On the one hand, the linguistic generalization is the result of a profound analysis of Russian phonology and morphology, which is a precondition for effective and theoretically grounded computational implementations. On the other hand, the need to find an effective implementational strategy can lead to interesting reanalyses of existing theoretical approaches.

When striving for the most efficient processing of language, grammar engineers can improve their systems in many different ways, including hardware development, software solutions like algorithm optimization and efficient grammar formalisms, and finally organizing the linguistic data by means of general, complete and, it is to be hoped, simple linguistic theories. In this paper we have focused on two of these steps. We demonstrated how to simplify linguistic analyses by looking at the data from a different perspective. We tried to justify our analyses by embedding them into a broader theory of Russian phonology and morphology. Additionally, we found a good compromise between network size and runtime efficency by combining two implementational strategies of the XEROX *xfst* tools into a hybrid approach. Another advantage of our implementation is that the data are processed in only one module, namely a simple FST morphology that can account for inflectional and derivational properties and for the interaction of grammatical and lexical meaning. One drawback is that the implementational strategy is restricted to the *xfst* toolbox. However, our unified approach facilitates the integration of the morphological output into the bigger grammar architecture of the Xerox Linguistic Environment (XLE) which has been developed for syntactic parsing, as described in Butt et al. (1999). It remains an interesting question whether similar phenomena can also be reanalyzed and/or implemented in this way.

We would like to conclude with a quotation from Karttunen (2003) where he describes the relationship between computational and theoretical linguistics. Computational knights have been constantly rejected by the Princess of Phonology and Morphology for more than three decades.

"This constant rejection of the most suitable suitor is puzzling. The Princess must have a vested interest in making simple things appear more complicated than they really are. The good news that the computational knights are trying to deliver is unwelcome. The Princess prefers the pretense that phonology/morphology is a profoundly complicated subject, shrouded by theories."

Maybe the next generation is not prejudiced in any way and open to fruitful interaction!

Acknowledgments

First of all we would like to thank Lauri Karttunen for his valuable ideas and the time he spent on our questions. Many thanks also go to Miriam Butt for her helpful comments and to Thomas Mayer for his help with earlier drafts of this version. Finally, we thank the two anonymous reviewers for their critical remarks which helped to improve our paper. Of course, we take the full responsibility for all errors.

References

Beesley, Kenneth R. 1998. Constraining separated morphotactic dependencies in constraining separated morphotactic dependencies in finite-state grammars. In L. Karttunen and K. Oflazer, eds., *FSMNLP'98. Proceedings of the International Workshop on Finite State Methods in Natural Language Processing*. Bilkent University Ankara, Turkey.

Beesley, Kenneth R. and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications,Stanford, California.

Breu, Walter. 1994. Interactions between lexical, temporal and aspectual meanings. *Studies in Language* 18(1):23–44.

Breu, Walter. 2000. Zur Position des Slavischen in einer Typologie des Verbalaspekts (Form, Funktion, Ebenenhierarchie und lexikalische Interaktion). In I. W. Breu, ed., *Probleme der Interaktion von Lexik und Aspekt (ILA)*, pages 21–54. Linguistische Arbeiten,Tübingen.

Butt, Miriam, Tracy Holloway King, María-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. CSLI Publications, Stanford, California.

Grimshaw, Jane. 1990. *Argument Structure*. MIT Press, Cambridge, Massachusetts.

Isačenko, A.V. 1995. *Die russische Sprache der Gegenwart – Formenlehre*. Max-Hueber-Verlag, München, 4th edn.

Jurafsky, Dan and James Martin. 2008. *Speech and Language Processing*. Prentice Hall Series in Artificial Intelligence, 2nd edn.

Karttunen, Lauri. 2003. Computing with realizational morphology. In A. Gelbukh, ed., *Computational linguistics and intelligent text processing. Lecture notes in computer science N 2588*, pages 205–216. Springer-Verlag, Berlin, Heidelberg.

Lahiri, Aditi and Vincent Evers. 1991. Palatalization and coronality. In C. Paradis and J.-F. Prunet, eds., *Phonetics and Phonology 2*, pages 79–100. Academic Press.

Lehmann, Volker. 1999. Aspekt. In H. Jachnow, ed., *Handbuch der sprachwissenschaftlichen Russistik und ihrer Grenzdisziplinen*, pages 214–242. Harrassowitz,Wiesbaden.

Plank, Frans. 1991. Inflection and derivation. *EUROTYP Working Papers* VII(10):1–28.

Roche, Emmanuel and Yves Schabes. 1997. Introduction. In E. Roche and Y. Schabes, eds., *Finite-State Language Processing*. MIT Press, Cambridge MA.

Sadler, Louisa, Andrew Spencer, and Marina Zaretskaya. 1996. A morphomic account of a syncretism in Russian deverbal nominalizations. In G. Booij and J. van Marle, eds., *Yearbook of Morphology*, pages 181–215. Kluwer Academic Publishers, Dordrecht.

Spencer, Andrew. 1998. *Morphological Theory*. Oxford: Blackwell Publishers.

Stump, Gregory. 1993. On rules of referral. *Language* 69:449–479.

Timberlake, Alan. 2004. *A Reference Grammar of Russian*. Cambridge University Press.