AN INVESTIGATION OF KAZAN TATAR MORPHOLOGY

_____

A Thesis

Presented to the

Faculty of

San Diego State University

_____

In Partial Fulfillment

of the Requirements for the Degree

Master of Arts

in

Linguistics

_____

by

Albina Raisovna Davliyeva

Spring 2011

SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Albina Raisovna Davliyeva:

An Investigation of Kazan Tatar Morphology

Robert Malouf, Chair
Department of Linguistics and Asian/Middle Eastern Languages

Robert Underhill
Department of Linguistics and Asian/Middle Eastern Languages

John Carroll
Department of Computer Science

NOV 22 2010
Approval Date

DEDICATION

To my grandmother Zakira who engrained in me the little Tatar that I know.

To my parents, Marsilia and Rais, who engrained in me the value of learning since the early age.

To my niece Diana and nephew Daniel, may you fall in love with the Tatar language.

ABSTRACT OF THE THESIS

An Investigation of Kazan Tatar Morphology
by
Albina Raisovna Davliyeva
Master of Arts in Linguistics
San Diego State University, 2011

This paper describes the morphological processor for parsing and generation of the Kazan Tatar language designed and implemented using XEROX finite-state tools. Kazan Tatar is one of the two official languages of the Republic of Tatarstan in the Russian Federation. It is characterized by rich and complex agglutinating morphology with phonologically determined allomorphy. This richness and complexity is defined by the number of suffixes, morpheme functions, and recursive word formation devices. Kazan Tatar grammar, including morphotactics and phonological processes, relevant to the project is discussed along with the implementation details.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

ABBREVIATIONS

Aff Affirmative

Comp Comparative

DB Derivational Boundary

FSA Finite-State Automation

FSN Finite-State Network

FST Finite-State Transducer

IF Intermediate Form

IPA International Phonetic Alphabet

KaTMorph Kazan Tatar Morphological Processor

N Noun

Neg Negative

NLP Natural Language Processing

NP Noun Phrase

Obj Object

POS Part of Speech

SF Surface Form

VP Verb Phrase

UF Underlying Form

XFST Xerox Finite-State Transducer

ACKNOWLEDGEMENTS

This project is carried out thanks to many people: my professors, my family, and my friends.

I would like to extend my deepest gratitude to my linguistics professors who have inspired me with their passion and knowledge of linguistics as well as their dedication to teaching (the names are listed alphabetically):

Jon Dayley

Jean Mark Gawron

Robert Malouf

Mary Ellen Rider

Gail Shuck

Robert Underhill

I owe my gratitude to my thesis committee members Robert Underhill, Robert Malouf, and John Carroll whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

I would also like to acknowledge my anthropology professors Robert McCarl and John Ziker for raising my interest in the cultural and social aspects of linguistics.

I would like to thank Hadi Karimov, Alfina Galiahmetova, and my parents who provided native speaker judgments. I am also thankful to Steve Milaskey, Karen Cook, Dale Spindler, Dave Walsh, and Jared Milaskey for proofreading the manuscript.

I would like to gratefully acknowledge my supervisor Mark Morsch for being accommodating with my school schedule.

I am thankful to Elvira, Steve, and Aliya for their moral support throughout my academic journey. I wish to thank my fellow graduate students Gina, Brandon, Dave, Rafi, and Paulo for providing a stimulating and fun environment to learn and grow.

Lastly, I thank all of those who supported me in any respect during the completion of this project.

CHAPTER 1

INTRODUCTION

OBJECTIVE

The goal of this project is to design and implement a Kazan Tatar Morphological Processor (KaTMorph) that can analyze a grammatical Kazan Tatar word by annotating its derivational and grammatical features, and can generate a well-formed Kazan Tatar word from a specified root and grammatical features. This processor is a finite-state "transducer that incorporates lexicon, morphotactic, and morphophonological alternations" (Beesley & Karttunen, 2003, p. xvi); it maps properly spelled words to morphosyntactic interpretation and vice versa. It is developed mostly for inflectional morphology of nouns and verbs; minimal derivational morphology is included in the source code to demonstrate word derivation in Kazan Tatar and to provide the framework for future development. KaTMorph should accept and produce the grammatical strings for the Kazan Tatar language, and reject and not generate ungrammatical strings. The task of the processor is to be able to take the input forms of the type listed in column 1 of Table 1, and to produce the output forms of the type listed in column 2.

To design a morphological processor, the following knowledge of the language structure is required (Jurafsky & Martin, 2008):

1. lexicon, which lists word roots and morphemes
2. morphotactic rules, which explain morpheme ordering within a word
3. phonological rules that affect orthography when morphemes combine

The choice of developing the morphological processor for Kazan Tatar using finite-state Networks (FSN) is due to the fact that concatenative morphology, characteristic of a Turkic language, can be elegantly described by finite automata (Trost, 2003). In addition, computing with FSN is attractive because they are computationally efficient, require relatively little memory for storage, are known for their accuracy, and their mechanism is language independent (Beesley & Karttunen, 2003). KaTMorph contains 28 noun, 16 verb, 6

Table 1. Morphological Analysis and Production

Input	Output	Gloss
Parsing		
miNa	min-Pron+Dat	to me
minem	min-Pron+P1-Sg+Poss	my
tapsInnar	tap-Verb+Imp+P3-PL	let them find
tapsIn	tap-Verb+Imp+P2-Sg	let him find
kolaklarnI	kolak-Noun+PL+Def-Acc	ears (Object)
kolaklar	kolak-Noun+PL+Nom	ears
matur	matur-Adj	beautiful
maturlIk	matur-Adj+^DB-Noun+Sg+Nom	beauty
Generation		
min-Pron+Dat	miNa	to me
min-Pron+P1-Sg+Poss	minem	my
tap-Verb+Imp+P3-PL	tapsInnar	let them find
tap-Verb+Imp+P2-Sg	tapsIn	let him find
kolak-Noun+PL+Def-Acc	kolaklarnI	ears (Object)
kolak-Noun+PL+Nom	kolaklar	ears
matur-Adj	matur	beautiful
matur-Adj+^DB-Noun+Sg+Nom	maturlIk	beauty

pronoun, and 3 adjective roots and generates an unlimited number of inflected word forms while occupying only 290 Kbytes[1] of memory and compiles in seconds.

PROJECT SIGNIFICANCE

The project significance is determined by the general importance of morphological knowledge for language processing by humans and computers, and by language specific considerations. In computational linguistics, the applications of morphological analyzers are vast. They can be used as stand-alone applications or as a preprocessing stage for other projects or systems in natural language processing (NLP) such as part of speech tagging, parsing, and translation.

A morphological analyzer can be used to investigate language morphology and syntax for linguistic research and for didactic purposes. Understanding word structure is important for comprehension of agglutinative languages because morphemes carry

[1]Interestingly, this project discussion paper occupies 796 Kbytes of memory.

information about grammatical relations of words in a sentence. For instance, understanding morphological word structure of languages like Kazan Tatar is necessary in order to disambiguate grammatical/thematic relations, especially in the sentences where understanding semantics is not enough. Sentence 1 is not thematically ambiguous[2]; it is obvious from the meaning of the words 'tea', 'mother', and 'drink' who the agent is even though morphologically both nouns are unmarked.

(1) Әни чәй эчә
Әni çəy eçə
Ani çAi eçA
Mother tea drinks
'Mother drinks tea'

In sentence 2, recognizing inflectional morphemes is important because we cannot rely on semantics or word order to distinguish the agent from the patient.

(2) Малай-ны кыз ярата
Malay- nı kız yarata
Malay- nI kIz yarata
Boy-Acc girl-Nom love-Pres-P3-Sg
'A/the girl loves the boy'

Knowing morphological processes can expand a language learner's lexicon acquisition and help in semantic, syntactic, and stylistic comprehension of a language (Sabitova, 2002).

Morphological processing is a necessary and useful stage in NLP systems. In spell checkers, for example, comparing an input word against a static list of words is impractical and impossible due to language productivity: it is not possible to manually list all morphological variants of every word[3] in an agglutinating language in advance (Jurafsky &

[2] Examples are listed in the following order: Cyrillic spelling, Latin spelling, xfst spelling (if different from Latin spelling), gloss; Latin spelling and gloss are provided for a Tatar word mentioned in the discussion; KaTMorph output examples are spelled with the alphabet used in the program (xfst column of Alphabet conversion Table 1). The meanings of grammatical tags can be found in Table 3. Tatar language data, unless specified otherwise, is self generated.

[3] A Kazan Tatar noun has 52 unique forms, not counting derivational forms.

Martin, 2008; Trost, 2003). However, a morphological analyzer, limited to mere word roots with morphotactic rules, is the better alternative to the static word lists. For the purposes of stemming, the process of stripping the word-end morphemes in Information Retrieval systems, rule based morphological analyzers tend to be more accurate than traditional stemmers (Tzoukerman, Klavans, & Strzalkowski, 2003). The morphological structure of a word produced by an analyzer carries syntactic and grammatical information about the word, information necessary for part of speech tagging and parsing sentences. This information is especially valuable for dependency parsing, which is used for synthetic and agglutinating languages.

Language choice for this project is determined by two factors: linguistic and geopolitical. Tatar has rich but regular morphology accompanied by various phonologic harmony rules, which makes it a suitable candidate for computing with finite-state networks.

Development of a morphological analyzer/generator requires implementation of morphological knowledge and accompanying phonological processes, which creates a digital record of a morphological model of a language. Despite its official status in the Republic of Tatarstan, the number of native speakers of Tatar is declining. This tool has the potential to contribute to language preservation and language learning as a stand-alone program or as a stage in various NLP systems for Kazan Tatar in the future.

There are a few Turkic languages for which computational morphological analyzers have been developed, including Turkish (Solak & Oflazer, 1992), Türkmen, and Crimean Tatar (Altintas & Cicekli, 2001). The automated morphological analyzer is an integral part of the Turkish spell-check application (Solak & Oflazer, 1992), the Turkish Dependency Treebank Tagging Annotation tool (Atalay, Oflazer, & Say, 2003), and the machine translation system between Turkish and Crimean Tatar (Altintas, 2001).

According to Suleymanov (2007), a two-level morphological analyzer was developed for Kazan Tatar by the Research Lab at Tatarstan Academy of Sciences jointly with Bilkent University; it is used for developing a machine translation system for the Turkish-Kazan Tatar language pair. However, to my knowledge, it appears that the analyzer is available only for the Kazan State University community.

KAZAN TATAR LANGUAGE

Kazan Tatar[4] belongs to the North-Western group of the Turkic languages (Gadzhieva, 1990). It is one of the 3 main dialects of Tatar, the other two being Western (Mishar), and Eastern (Siberian). Since 1992 it is one of the two official languages in the Republic of Tatarstan, Russian Federation. In the Russian Federation, the term 'Tatar' refers to Kazan Tatar. Throughout this discussion this term will also be in reference to Kazan Tatar.

According to the Russian census of 2002, there are over 5.5 million Tatars living in the Russian Federation (Federal State Statistics Service, 2004), although it would be inaccurate to assume that all are Tatar speakers. People that identify themselves of Tatar ethnicity are not necessarily the native speakers of Tatar. According to the demographic report by Tatarstan Academy of Sciences, demographic reasons such as urbanization, low birth rate, intermixed marriages, and the fact that Tatar is not a prestige language, cause the numbers of native speakers of Tatar to decline (Isxakov, 2007*)*. To make matters worse, the medium of instruction in the majority of Tatarstan educational institutions is Russian; so only 53% of Tatar children study in their native language (Svechnikov & Sergeeva, 2008).

Throughout history, the writing system of the Tatar language used several alphabets: Arabic script was used until 1927, followed by Latin script until 1939 (*Tatar language*, 2009); currently the writing system is based on the Cyrillic alphabet. Alphabet Transliteration Table 2 lists the Latin and Cyrillic alphabets along with the phonetic alphabet which represents Kazan Tatar pronunciation. The 'XFST' column lists the modified alphabet used in KaTMorph.

In Tatar, the basic word order is Subject Object Verb. However, non-subject NPs can precede subjects for pragmatic reasons, such as expressing topicality. Adverbial phrases of place and time can be sentence initial; nominal modifiers precede their heads. In literary Tatar, the predicate is sentence final[5] and can be expressed by a verb, noun, adjective, or numeral. Verb and Agent agree in person and number. In copular sentences the copular verb is present in the future and past tense, and obligatorily absent in the present tense. Sentence 3

[4] Kazan is situated on the Volga River. Kazan Tatar is also referred to as Volga Tatar.

[5] Prescriptive Tatar grammar prohibits verbs in non-final position. To determine whether verbs are always sentence final, spoken language data needs to be gathered and examined.

Table 2. Alphabet Transliteration

Latin[6]	Cyrillic	XFST	IPA
A	А	a	ɑ
Ә	Ә	A	æ
B	Б	b	b
C	Җ	c	ʒ
Ç	Ч	C	tʃ
D	Д	d	d
E	Е	e	e
F	Ф	f	f
G	Г	g	g,G
H	Һ	h	h
I	Ы	I	ɨ
İ	И	i	i
J	Ж	j	ʒ
K	К	k	k,q
L	Л	l	l
M	М	m	m
N	Н	n	n
Ң	Ң	N	ŋ
O	О	o	o
Ө	Ө	O	ø
P	П	p	p
R	Р	r	r
S	С	s	s
Ş	Ш	S	ʃ
T	Т	t	t
U	У	u	u
Ü	Ү	U	y
V	В	v	v
X	Х	x	x
Y	Й	y	j
Z	З	z	z

[6] Source for the "Latin" and "Cyrillic" columns is 'Republic of Tatarstan' (2010).

has the canonical word order; and sentences 4, 5, and 6 are the examples of the clauses without the copular.

(3) Син китап-ны укы-дың

sin kitap-nɪ ukɪ-dɪŋ

sin kitap-nI ukI-dIN.

You book-Sg-Def-Acc read-PastDef-P2-Sg

'You read the book.'

(4) Бу укутучы

Bu ukutuçɪ

Bu ukutuCI

This teacher-Sg-Nom

'This is a teacher.'

(5) Бу күлмәк иске

Bu külmək iske

Bu kUlmAk iske

This shirt-Sg-Nom old

'This shirt is old.'

(6) Балалары өч-әү (adopted from Nasibullina, 2008)

Balalarɪ өç-әü

BalalarI Oç-AU

Child-PL-Poss-P3-Nom three-Pron-Coll

Their children three

'They/She/He have/has three children'

Due to the subject-verb concord in person and number, subjects in a Tatar sentence are optional. Both sentences 7 and 8 are grammatical; the subject in sentence 7 is inferred from the verb's suffix for person and number and is the same as in sentence 8. Even though these two sentences are not pragmatically equivalent, they are semantically and thematically equivalent.

(7) Китапны укыдым
Kitap- nɪ ukɪ-dɪ-m
Kitap-nI ukI-dI-m
book-Acc-Def read-PastPerf-P1-Sg
'I read the book'

(8) Мин китап-ны укыдым
Min kitap- nɪ ukɪ-dɪ-m
Min kitap- nI ukI-dI-m
I-Nom book-Acc-Def read-PastPerf-P1-Sg
'I read the book'

Tatar Phonetic Model

The Tatar alphabet consists of 9 vowels and 22 consonants[7]. For the purpose of this processor, vowels are subdivided into front and back vowels (in Tatar, corresponding terminology is 'soft' and 'hard' vowels). Tables 3 and 4 illustrate transliteration between the Latin vowels and the corresponding vowels used in the program. The vowels in the tables are lined up with their corresponding pair. For example, the front 'ü' and the back 'u' form a pair. It should be noted that in suffixes, the front vowel 'e' corresponds to the back vowel 'ɪ'. In addition, the vowel 'ə' is the orthographic representation of the sound [æ], and should not be confused with the schwa sound.

Table 3. Front Vowels

Latin	İ, i	Ü, ü	Ө, ө	Ә, ə	E, e
xfst	i	U	O	A	e

Table 4. Back Vowels

Latin	I, ɪ	U,u	O,o	A,a
xfst	I	u	o	a

[7] Table 1 lists the Tatar alphabet transliteration.

The Latin base spelling of the Tatar consonants and their xfst representation is presented in Table 5.

Table 5. Tatar Consonants

Latin	XFST		Latin	XFST
B, b	b		N, n	n
C, c	c		Ң, ң	N
Ç, ç	C		P, p	p
D, d	d		R, r	r
F, f	f		S, s	s
G, g	g		Ş, ş	S
H, h	h		T, t	t
J, j	j		V, v	v
K, k	k		X, x	x
L, l	l		Y, y	y
M, m	m		Z, z	z

The main classification of the Tatar consonants relevant for the development of KaTMorph is their division into voiced and voiceless consonants. The consonants used in the processor and throughout this discussion are:

Voiced Consonants: [b | c | d | v | g | z | j | m | n | N | y | l | r]

Voiceless Consonants: [p | C | t | f | k | s | S | h | x]

In Tatar, every morpheme has several variants (allomorphs) which are phonologically conditioned, with the exceptions of personal pronouns *min* 'I' and *sin* 'you', which are non-harmonic in dative case. The maximum number of possible allomorphs per morpheme is six. Allomorphs are derived from the underlying morpheme for the following reasons: (a) vowel harmony; (b) nasal assimilation; (c) assimilation by the [voice] feature. The underlying form (UF) is the starting point of the morphological derivation; the orthographic form after applying phonological rules is the surface form (SF). It should be noted that the listed phonological processes are not all-inclusive; they are selected based on their relevance for

this project. These processes are chosen because they change the orthographic form of a word.

(a) Vowel Harmony can be defined as the process of vowel assimilation to the preceding vowel's [back] feature. As a consequence, a Tatar word cannot contain both front and back vowels, with the exceptions of foreign borrowings such as kitap 'book', compound words such as *өç-poçmak* 'three-corners'[8], and previously mentioned personal pronouns, which are non-harmonic. In borrowings and compounds the [back] feature of a suffix vowel is chosen based on the vowel quality of the preceding syllable. Compare the word pairs in Table 6 where ungrammatical words are marked with an asterisk (*):

Table 6. Examples of Vowel Harmony Rules

	Back	Front	Front-Back
Back	bar-a 'goes'	*bər-a	kitap-lar 'books'
Front	*bar-ə	бəp-ə 'beats'	*kitap-lər

(b) Consonant Nasal Assimilation in Tatar is progressive: consonants 'l' and 'd' copy the [+nasal] feature of a preceding consonant. Examples 9 and 10 demonstrate how the consonants in underlying forms of plural suffix –lar and ablative case suffix -dan assimilate to the preceding nasal consonant and become –nar and –nan respectively.

(9) UF: tun-lar
SF: tun-nar
furcoat-PL-Nom
'furcoats'

(10) UF: tun-dan
SF: tun-nan
furcoat-Sg-Abl
'from furcoat'

[8] □ç*poçmak* 'three-corners' ≈ 'triangle' is the traditional Tatar pie made of meat and potatoes

(c) Consonant Voice Assimilation results in a change of the [voice] feature which affects the consonants of both bound and free morphemes. Example 11 illustrates how by progressive assimilation, –d or –g at the beginning of the inflectional morphemes become voiceless after voiceless consonants (Nasibullina, 2008).

(11)	SF: kit-de	kolak-ga
	UF: kit-te	kolak-ka
	Leave-PastDef-P3-Sg	ear-Sg-Dat
	'he/she left'	'to a/the ear'

The stem final consonants –p or –k become voiced in the intervocalic position, where the right context vowel is a part of an inflectional morpheme (Nasibullina, 2008). Example 12 demonstrates that root-final voiceless consonants become voiced between vowels[9]:

(12)	UF: kolak-ɪm	tap-ar
	SF: kolag-ɪm	tab-ar
	ear-Sg-Poss-P1-Sg	find-FutIndef-P3-Sg
	'my ear'	'he/she will find'

Tatar Morphology

Morphology is the branch of linguistics that studies word structure. Morphological knowledge is the knowledge of morphemes, and knowledge of the rules to combine them; these rules are called morphotactic rules. A morpheme is the smallest meaningful unit of language expressing semantic concepts or grammatical features (Fromkin, Rodman, & Hyams, 2003). Morphological analysis is the process of identifying morphemes in a given word. Morphemes that can be words are free morphemes, and the ones that must attach to stems are bound morphemes. In Tatar, bound morphemes are added to a word by suffixing. To form a new word with a new meaning a derivational affix is added. To change the

[9] Other voiceless consonants, for example 't', do not become voiced:

UF: kibet-em

SF: kibet-em

'my store'

grammatical meaning of a word inflectional morphemes are used; they don't change the syntactic category of words, but mark their properties such as case, gender, number, and tense (Fromkin et al., 2003). Morphologically complex words consist of a free morpheme, or root, which carries the main lexical content, and at least one derivational morpheme; thus a new stem can be formed. By adding more affixes to the stem, yet more complex stems can be formed (Fromkin et al., 2003). A bound morpheme can have several realizations which are in complementary distribution to each other; they constitute a set of allomorphs.

As an agglutinative language, Tatar has rich and complex morphology. This richness and complexity is defined by the number of the underlying suffixes, allomorphs, morpheme functions, and recursive word formation devices. Tatar suffixes are used to derive new words, to express grammatical relations or features, and to express pragmatic meaning (Ganiev, 2000). There is an arsenal of over 300 derivational suffixes (Ganiev, 2000), and word formation is recursive, so that a derived word can be a new stem for a new derivation.

Tatar bound morphemes are suffixes: they attach after a root or a stem. Each part of speech has its own set of suffixes, and each suffix has a set of phonologically determined allomorphs. The inflectional morphemes for nouns include case, number, and possessive features; for verbs, they are tense, aspect, mood, person, number, and negation, just to name a few (Ganiev, 2000). Grammatical details for each part of speech are provided in the corresponding sections of this discussion.

It should be noted that there is one morpheme that can attach to any part of speech—it is the 'yes-no question' morpheme. The polar question is formed by adding the underlying interrogative suffix –mɪ to a predicate or to any word in a sentence for 'contrastive purposes' (Underhill, 1986, p.59). Examples 13 illustrate the question formation for major parts of speech (for grammatical tag meanings, see Morphological Feature Tags, Table 7).

(13)

(a)

Син китап-ны укы-дың-мы?

sin kitap-nɪ ukɪ-dɪŋ-mɪ?

sin kitap-nI ukI-dIN-mI?

You book-Sg-Def-Acc read-PastDef-P2-Sg-Ques

'Did you read the book?'

Table 7. Morphological Feature Tags

Adj	adjective
Noun	noun
Verb	verb
Pron	pronoun
+	morpheme boundary
^DB-Adj	adjective derived from another word
^DB-Noun	noun derived from another word
P1	1st person
P2	2nd person
P3	3rd person
Sg	singular
PL	plural
Indef	indefinite
Def	definite
Nom	nominative
Acc	accusative
Gen	genitive
Dat	dative
Loc	locative
Abl	ablative
Abl2	alternative ablative
Poss	possessive-personal
Coll	collective
Pres	present
PastIndef	past indefinite
PastDef	past definite
FutIndef	future indefinite
FutDef	future definite
Imp	imperative
Neg	negation
Ques	interrogative

(b)

Бу укутучы-мы?

Bu ukutuçı-mı?

Bu ukutuCI-mI?

This teacher-Sg-Nom-Ques

‘Is this a teacher?’

(c)

Бу күлмәк иске-ме?

Bu külmək iske-me?

Bu kUlmAk iske-me?

This shirt-Sg-Nom old-Ques

‘Is this shirt old?’

(d)

Мин-ме?

Min-me?

I-Ques

‘me?’

(e)

Бу-мы?

Bu-mı?

Bu-mI?

This-Ques

‘This?’

(f)

Китап-ны-мы?

Kitap-nı-mı?

Kitap-nI-mI?

Book-Acc-Def-Ques

‘The book?’(Obj)

(g)

Барган-мы?

Bar-gan-mɪ?

Bar-gan-mI?

Go-PastIndef-P3-Sg-Ques

(he/she) went?

‘Did he/she go?’

(h)

Шат- сыз- лык- мы?

şat- sɪz- lɪk- mɪ?

Sat- sIz- lIk- mI?

Joyful-^DB-Adj-^DB-Noun-Ques

Joyful-without-N?

‘Unjoyfulness?’

MORPHOTACTIC RULES

The order of suffixes in a Tatar word is fixed. Tatar nouns, verbs, pronouns and adjectives have their own set of morphotactic rules; however, these rules have the following commonalities. Derivational suffixes immediately follow the root or another derivational suffix, inflectional morphemes come next, and the interrogative morpheme is always word final. Because word derivation is recursive, there could be more than one derivational suffix in a word. The morphotactic rules used in the processor are summarized in Figure 1 (note that derivational suffixes for verbs are not included in the current version of the program, but listed here only for illustrative purpose).

Noun Root + Derivational Suffix + Plural Suffix + Personal Suffix + Case + Interrogative Suffix Verb Root + Derivational Suffix + Negation + Tense + Personal Ending + Interrogative Suffix Pronoun + Case + Interrogative Suffix Adjective Root + Derivational Suffix + Interrogative Suffix

Figure 1. Morphotactic rules.

RECURSIVE CONCATENATION

My present work has a focus on Tatar inflectional morphology with a goal to provide morphological analysis of nouns, personal pronouns, adjectives, and verbs. However, I included the minimal derivational morphology in the lexicon in order to demonstrate the recursive behavior of word formation and to provide a framework for future development.

The noun lexicon has a finite number of morphemes: 12 Inflectional, and 8 derivational; yet the number of words that can be generated by the grammar is unlimited, as is the word length. The grammar defined for nouns and adjectives is recursive, which allows the generation of an infinite number of words. The circularity results from the loops in the noun lexicon, specifically due to the derivational rules for noun and adjective formation. Figure 2 is an illustration of the recursive behavior of the word generation from the root *süz* 'word' where new nouns and adjectives are formed by reapplying derivational morphemes to the newly formed stems.

s*üz*
s*üz*-Noun+Sg+Nom
'word'

süz-le
s*üz*-Noun+^DB-Adj
'talkative' = verbose

süz-le-lek
s*üz*-Noun+^DB-Adj+^DB-Noun+Sg+Nom
'quality of being talkative' = verbosity

süz-le-lek-sez
s*üz*-Noun+^DB-Adj+^DB-Noun+^DB-Adj
'the one without the quality of being talkative' = the one without verbosity = taciturn (the shorter version, s*üzsez*, means 'speechless')

süz-le-lek-sez-lek
s*üz*-Noun+^DB-Adj+^DB-Noun+^DB-Adj+^DB-Noun+Sg+Nom
'the quality of being without the quality of being talkative' = taciturnity

Figure 2. Tatar word formation.

Even though these words are possible in the language, the comprehension of such complex derived words gets more challenging as the word 'grows' longer. If necessary, it is possible to restrict such recursive generation by applying a 'filter' on top of the lexicon

(Beesley & Karttunen, 2003). This filter can eliminate the undesired strings based on the limit we set on the number of unique derivational morphemes a word can have. The true challenge is determining this limit. Another option is to set semantic restrictions, which would allow only certain roots to be the stems for new derivations. To implement such restriction, the most productive suffixes should be factored out from the set of the derivational suffixes. This would allow all nouns or adjectives to get productive derivational suffixes, and only a limited number of nouns or adjectives to get the non-productive ones[10]. Note, however, that such restriction is not implemented in the current version of the processor.

The following sections present an overview of morphological aspects of Tatar language relevant to the design of this morphological processor.

NOUN

Grammatical categories of nouns are number, possession, and case; they are expressed by inflectional suffixes. The underlying forms of inflectional and derivational morphemes have multiple allomorphs conditioned by vowel harmony rules and nasal or voice assimilation as discussed in the Tatar Phonetic Model section.

In a noun, suffixes are sequenced as follows: Noun Root + Derivational Suffix+ Inflectional Suffix + Interrogative Suffix. The inflectional morphemes for nouns are optional, but they do have to attach in the following order: the plural suffix, a personal suffix, and a case suffix. Thus, the morphological structure of a noun is schematically represented as follows: Noun Root + Derivational Suffix + Plural Suffix + Personal Suffix + Case Suffix + Interrogative Suffix.

The derivational suffixes form a new noun with new meaning by appending to noun or adjective roots or stems. Table 8 demonstrates the most productive Tatar suffixes for noun formation included in the KaTMorph lexicon (Nasibullina, 2008).

Due to the recursive noun formation, the maximum number of morphemes a noun can have is unlimited, but the number of non-derivational suffixes is limited to four: plural, personal, case, and question. Example 14 demonstrates how the noun is composed by the

[10] See Beesley & Karttunen (2003, p.245) for details.

Table 8. Noun Derivational Suffixes

stem	suffix	meaning	example
noun	ç□	person ‘doer’	*süz-lek-çe* ‘lexicographer’
noun, adj	l□k	abstract noun	*süz-lek* ‘dictionary’ ; *şat-l□k* ‘joy’
noun	l□	person from	*kazan-l□* ‘kazanien’

final noun rule. The noun is derived from an adjective and is in plural form functioning as a direct object in a question form:

(14) Шат- лык- лар- ы- ны- мы

Sat- lIk- lar- I- nI- mI

Sat-Adj+^DB-Noun+ PL+ Poss+P3+ Def-Acc+ Ques

Joyous+Noun+PL+Poss+P3+Def+Acc+Ques

‘Their joys-Acc?’

Next, let me present an overview of Tatar noun grammar reflected in the lexicon of the Morphological Processor.

Number

Tatar nominals have the grammatical feature of number. The plural form is marked by the underlying suffix –lar attached to the noun root. This underlying form changes according to the morphophonological rules. First, if the noun ends with the nasal consonant [n, ŋ, m], then –l becomes –n. Second, -a- changes to -ə- when the base noun has a front vowel in the final syllable. Table 9 demonstrates the surface forms of all 4 possible forms of the plural morpheme.

Table 9. Tatar Noun Singular and Plural Forms

Sg Noun	Singular	Plural
[+back] vowels	bala ‘child’	bala-lar ‘children’
[-back] vowels	keşe ‘person’	keşe-lər ‘people’
nasal,[+back] vowels	uram ‘street’	uram-nar ‘streets’
nasal, [+back] vowel	kөn ‘day’	kөn-nər ‘days’

It should be noted that the plurality feature can also be expressed syntactically, by means of quantative pronouns or numerals with the noun in the unmarked form; the processor will mark the nouns in such expressions as morphologically singular:

Күп бала
küp bala
kUp bala
Many child-Sg-Nom
'many children'

Биш бала
Biş bala
Five child-Sg-Nom
'Five children'

Personal Suffixes (Possessive)

There is a group of morphemes in Tatar indicating possession; there are 6 of them distinguishing person and number of a possessor. These suffixes attach to the noun roots referring to possessed thing(s) or person(s). These morphemes can attach either to a singular or plural form of a noun. The surface forms of each morpheme are governed by phonological processes discussed in the Phonetic Model Section. Table 10 lists the underlying personal suffixes and Table 11 lists the declension of the noun 'aunt' by possession in nominative case.

Table 10. Tatar Personal Suffixes

Sg Possessor	post consonant or *ü/u*	post vowels
1	ɪm	m
2	ɪŋ	ŋ
3	ɪ	sɪ
PL Posessors		
1	ɪbɪz	bɪz
2	ɪgɪz	gɪz
3	ɪ	sɪ

Table 11. Noun Declension by Possesion

Sg	apa 'aunt'	apa-lar 'aunts'
1	apa-m 'my aunt'	apa-lar-ɪm 'my aunts'
2	apa-ŋ 'your aunt'	apa-lar-ɪŋ 'your aunts'
3	apa-sɪ 'his/her aunt'	apa-lar-ɪ 'his/her aunts'
PL		
1	apa-bɪz 'our aunt'	apa-lar-ɪbɪz 'our aunts'
2	apa-gɪz 'Your aunt'	apa-lar-ɪgɪz 'Your aunts'
3	apa-sɪ 'their aunt'	apa-lar-ɪ 'their aunts'

Case

Case is used to express grammatical relations between words in a sentence. The case feature in Tatar is expressed syntactically by means of postpositions and morphologically by means of suffixes. Current discussion is concerned with morphological cases only. Most Tatar linguists agree that the Tatar noun has at least six morphological cases (Ganiev, 2000). These cases are nominative, genitive, dative, accusative, ablative, and locative. Table 12 is a declension table for Tatar nouns.

Table 12. Case Paradigm for Nouns

Case	alma ‘apple’	et ‘dog’
Nominative	alma	et
Genitive	alma-nɪŋ	et-neŋ
Dative	alma-ga	et-kə
Accusative Definite	alma-nɪ	et-ne
Accusative Indefinite	alma	et
Ablative	alma-dan	et-tən
Locative	alma-da	et-tə

The dictionary form of a noun is listed in the morphologically unmarked, nominative singular form. It has the functions of (a) the nominative case, (b) the indefinite accusative case, and (c) the indefinite genitive case:

a) Nominative case
    1) A noun functions as a subject of a sentence. In (15) the noun ‘woman’ is the subject:

(15) Бу апа бик матур.

Bu apa bik matur.

This woman very beautiful.

'This woman is very beautiful'

2) A noun functions as a nominal predicate. In (16) 'student' is the nominal predicate.

(16)Бу кыз студент

Bu kız student

Bu kIz student

this girl student

'this girl is a student'

3) A noun functions as a vocative, as the noun *iptəşlər* 'friends' in sentence 17.

(17) Җырлыйк әле, иптәшләр. (Tatar folk song)

cırlıyk əle, iptəş-lər.

cIrlIyk Ale, iptAS-lAr.

Sing-Imp Imp-Part friend-Pl

'Let us sing, friends'

b) Indefinite accusative case

A noun in this form functions as an unspecified or nonreferential direct object of a verb. In (18) the noun 'book' is the morphologically unmarked direct object:

(18) Мин китап укыйм

Min kitap ukıym

Min kitap ukIym

I -Nom book-Indef-Acc read-Pres-P1-Sg

'I am reading a book'

c) Indefinite genitive case (Ganiev, 2000).

A noun in this form is nonreferential; 'dog' in (19) is such a noun.

(19) эт койрыг-ы

et koyrıg-ı

et koyrIg-I

dog-Nom tail-Sg-Poss-P3-Sg

dog's tail

The accusative case marks a uniquely identifiable direct object of a verb by adding the underlying suffix -nɪ. The direct object in (20) refers to the specific book.

(20) Мин китап-ны укыйм

Min kitap-nɪ ukɪym

Min kitap-nI ukIym

I book-Acc-Def read-Pres-P1-Sg

'I am reading the book'

A noun in the dative case form has the underlying ending –ga and expresses time, location, goal, or an indirect object of a verb (Zakiev & Ramazanova, 2002). In example (21) the indirect object is marked with the dative case.

(21) Ул малай-га китап бирде

ul malai-ga kitap bir-de

He boy-Dat book-Acc-Indef give-PastDef-P3-Sg

'He gave a boy a book'

The locative case, as the name suggests, expresses the location of the action or event: өydə 'at home' (22). The same case is used to express the place in time: noyabrda 'in November'.

(22) Өй-дә ул миң-а китап бирде

өy-də ul miŋ-a kitap bir-de

Oy-dA ul miNa kitap bir-de

Home-Loc he I-Dat book-Acc-Indef give-PastDef-P3-Sg

'At home, he gave me a book'

The literal translation of the ablative case from Tatar is a 'point of departure'. It is formed by adding the underlying –dan to the noun roots and it carries the functions of expressing

(a) 'the place from which' or 'the place through which' (Lewis,1967)

өй-дән

өy-dən

Oy-dAn

Home-Abl

‘from home’

ишек-тән

işek-tən

iSek-tAn

door-Abl

‘through door’

(b) ‘reason for a state’

шатлык-тан

şatlık-tan

SatlIk-tan

Joy-Abl

‘from (because of) joy’

(c) in comparative constructions

Казан Уруссу-дан зур-рак

Kazan Urussu-dan zur-rak

Kazan-Nom Urussu-Abl big-Comp

‘Kazan is bigger than Urussu’

(d) the material from which something is made

Бу алка-лар алтын-нан яса-л-ган

Bu alka-lar altın-nan yasa-l-gan

Bu alka-lar altIn-nan yasa-l-gan

This earring-PL gold-Abl make-Passive-PastIndef-P3-Sg

‘These earrings are made of gold’

The genitive case underlying morpheme -nıŋ attaches to a noun referring to a possessor (qualifying noun). It can attach to the root (23) or to the stem marked with personal suffix (24).

(23) Апа-ның өй-е

apa-nıŋ өy-e

apa-nIN Oy-e

aunt-Gen house-Sg-Poss-P3-Sg

‘aunt’s house’

(24) Апа-лар-ыгыз- ның өй-е

apa- lar- ɪgɪz- nɪŋ өy-e

apa- lar- IgIz- nIN Oy-e

aunt-PL- Poss-P2-PL-Gen house-Sg-Poss-P3-PL

aunts-yours-theirs house

‘your aunts’ house’

Definiteness

In Tatar, definiteness is expressed by morphological case alternation, so the NP in the same grammatical function can have different case markings (Aissen, 1999). In Tatar, Nominative-Accusative and Nominative-Genitive alternations are used to express the notions of definiteness, referentiality, and specificity. Morphologically unmarked forms are for indefinite or non-referential nouns, and the accusative and genitive suffixes are used to mark definite nouns.

In canonical, nominative-accusative transitive constructions, the morphological realization of the direct object case is dependent on definiteness: an indefinite direct object is morphologically unmarked, and a definite direct object has the accusative suffix. Examples 25 and 26 illustrate the contrast in definiteness via direct object alternation: *alma* ‘apple’ functioning as the direct object is in the nominative form to express indefinite (25), and has the accusative ending –nɪ to express the definite direct object (26).

(25) Ул алма ашады

ul alma-NULL aşadɪ

ul alma-NULL aSadI

She/He-Nom apple-NULL-Indef eat-PastDef-P3-Sg

‘She/He ate an apple[11].’

‘She/He ate apples.’

[11]Morphologically unmarked objects are number neutral (R, Underhill, personal communication, November 2, 2010)

(26) Ул алма-ны ашады

ul alma-nɪ aşadɪ

ul alma-nI aSadI

She/He-Nom apple-Acc-Def eat-PastDef-P3-Sg

'He ate the apple.'

In possessive constructions, the modifying noun has a function of the genitive case; its morphological realization is dependent on definiteness or referentiality of the possessor to which the noun is referring. NP constructions in 27 and 28 are examples of the Genitive-Nominative alternation. In (27) 'dog' is a generic, nonreferential noun, and 'mother' is a definite noun (28).

(27) эт койры-гы

et koyrɪg-ɪ

et koyrIg-I

dog-Nom tail-Sg-Poss-P3-Sg

'dog's tail'

(28) әни-нең сумкас-ы

әni-neŋ sumka-sɪ

Ani-neN sumka-sI

mother-Gen purse-Poss-P3-Sg

'mother's purse'

In NP (27) the phrase-final head noun is marked with the possessive suffix of the 3rd person and the modifying nonreferential noun is in nominative form. These types of constructions are best described as Noun-Noun compounds. The relationship between the two nouns is not possessive, but merely 'qualificatory' (Lewis, 1967). In NP (28) the head noun is marked with the possessive suffix in the 3rd person, and a referential possessor has the genitive marker (Nasibullina, 2008).

In addition to case marking, direct object definiteness can be expressed by determiners.

1) With definite nouns marked by the overt accusative ending
    a. Optional determiner *bu* 'this/these' can be used:

bu alma-nɪ

this apple-Acc

‘this apple’(Obj)

b. Optional determiner *ber* ‘one’ is used to express the quantity:

Ber alma-nɪ

one apple-Def-Acc

‘one (particular) apple’(Obj)

c. Optional possessive pronoun can be used:

Min-em xatɪn-nɪ

my wife-Def-Acc[12]

‘my wife’(Obj)

2) With indefinite nouns without case marking the optional determiner *ber* ‘one/a’ can be used:

ber alma

one apple-Indef-Acc

‘one/an apple’(Obj)

It should be noted that KaTMorph will return two analyses for the noun in the unmarked form. For example, there are two parses for the word *eş* ‘work’:

eS-Noun+Sg+Nom

eS-Noun+Sg+Indef-Acc

Nouns marked by personal suffixes are definite by definition; therefore they cannot be analyzed as indefinite accusative. This processor will produce an unambiguous analysis for such forms:

eş-ebez

eS-Noun+Sg+Poss+P1+PL+Nom

‘our work’

To summarize, Tatar noun paradigms defined in the processor’s lexicon include (structure is adopted from Underhill (1986)):

1. Noun Root
2. Derivational Suffixes

[12] the same meaning ‘my wife-Acc’ can be expressed by means of a personal morpheme -ɪm: xatɪn-ɪm-nɪ. Yet, one more way to express the same semantic meaning is *minem xatɪn-ɪm-nɪ*.

3. Plural
4. Personal Suffixes
5. Case

The inflectional paradigm for the word *apa* 'aunt' is presented in the series of Table 13a-d. Note that there are no accusative indefinite forms for the nouns marked with possessive suffixes.

Table 13a. Declension of *apa*

	aunt	aunts
	Sg	PL
Nom	apa	apalar
Gen	apanɪŋ	apalarnɪŋ
Dat	apaga	apalarga
Acc-Def	apanɪ	apalarnɪ
Acc-Indef	apa	apalar
Abl	apadan	apalardan
Loc	apada	apalarda

Table 13b. Declension of *apa*, 1st Person Possessor

	my aunt	my aunts	our aunt	our aunts
	Sg Poss P1 Sg	PL Poss P1 Sg	Sg Poss P1 PL	PL Poss P1 PL
Nom	apam	apalarɪm	apabɪz	apalarɪbɪz
Gen	apamnɪŋ	apalarɪmnɪŋ	apabɪznɪŋ	apalarɪbɪznɪŋ
Dat	apama	apalarɪma	apabɪzga	apalarɪbɪzga
Acc-Def	apamnɪ	apalarɪmnɪ	apabɪznɪ	apalarɪbɪznɪ
Acc-Indef	--	--	--	--
Abl	apamnan	apalarɪmnan	apabɪzdan	apalarɪbɪzdan
Loc	apamda	apalarɪmda	apabɪzda	apalarɪbɪzda

Table 13c. Declension of *apa*, 2nd Person Possessor

	your-Sg aunt	your-Sg aunts	Your-Pl aunt	Your-Pl aunts
	Sg Poss P2 Sg	PL Poss P2 Sg	Sg Poss P2 PL	PL Poss P2 PL
Nom	apaŋ	apalarɪŋ	apagɪz	apalarɪgɪz
Gen	apaŋɪŋ	apalarɪŋnɪŋ	apagɪznɪŋ	apalarɪgɪznɪŋ
Dat	apaŋa	apalarɪŋa	apagɪzga	apalarɪgɪzga
Acc-Def	apaŋnɪ	apalarɪŋnɪ	apagɪznɪ	apalarɪgɪznɪ
Acc-Indef	--	--	--	--
Abl	apaŋnan	apalarɪŋnan	apagɪzdan	apalarɪgɪzdan
Loc	apaŋda	apalarɪŋda	apagɪzda	apalarɪgɪzda

Table 13d. Declension of *apa*, 3rd Person Possessor

	his/her/their aunt	his/her/their aunts
	Sg Poss P3	PL Poss P3
Nom	apasɪ	apalarɪ
Genitive	apasɪnɪŋ	apalarɪnɪŋ
Dat	apasɪna	apalarɪna
Acc-Def	apasɪn	apalarɪn
Acc-Indef	--	--
Abl	apasɪnnan	apalarɪnnan
Loc	apasɪnda	apalarɪnda

VERB

Next, let me present an overview of Tatar verb morphology relevant to this project. The dictionary form of a verb is Infinitive, for example, *bararga* 'to go'. The root of a verb is its imperative 2nd person singular form such as *bar* 'go'. It is verb roots, not dictionary forms, that are lexical entries in this morphological processor (the reasoning for this choice is discussed in the Lexicon section of the Methodology Section).

Verb grammatical features defined in this program are aspect, tense, person, number and negation (Ganiev, 2000); they are expressed via morphology. By the morphotactic rules, an optional derivational morpheme[13] can be added to the stem, followed by an optional negation morpheme, a tense morpheme, a personal suffix, and ending with an optional interrogative suffix; schematically these rules are represented as follows: Verb Root + Derivational Suffix + Negation + Tense + Personal Ending + Interrogative Suffix.
Verb suffixes representing its grammatical features will be discussed in the remaining part of this section and they are:

1. Tense: Present, Past Indefinite, Past Definite, Future Indefinite, Future Definite
2. Personal: Person and Number

Negation

To negate a sentence with a verbal predicate, the negation morpheme is added after the verb root before the tense or mood suffix. Examples 29 and 30 illustrate the negated sentences in the past definite tense and the imperative mood.

[13] The verb derivational morphology is not implemented in the current version of this program.

(29) Син китапны укы-ма-дың

sin kitapnɪ ukɪ-ma-dɪŋ

sin kitapnI ukI-ma-dIN

You book-Sg-Def-Acc read-not-PastDef-P2-Sg

'You did not read the book'

(30) Кил- мә- сен

Kil- mə- sen

Kil- mA- sen

Come-not- Imp-P3-Sg

'Let him/her not come'

The negation suffix has 3 allomorphs: -ma, -mə, and -m. Verbs in the future and past tenses and the imperative mood get –ma/-mə[14] and the present tense verbs get the suffix -m (Nasibullina, 2008). Table 14 illustrates the negation allomorph distribution for the 3rd person Singular.

Table 14. Verb Negation

	Imperative	Present	Past Indef	Future Indef
'come'	kil-mə-sen	kil-m-i	kil-mə-gən	kil-mə-s
'go'	bar-ma-sɪn	bar-m-ɪy	bar-ma-gan	bar-ma-s

A verb in the negative form has the same tense morphemes as its positive counterpart, with the exception of the present and future indefinite tense. Table 15 has both affirmative and negated verb forms for *bar* 'go' conjugated by tense for the 3rd person singular.

Table 15. Verb Conjugation by Tense for the 3rd Person Singular

	Affirmative	Negative
Imperative	bar-sɪn	bar-ma-sɪn
Present	bar-a	bar-m-ɪy
Past Indef	bar-gan	bar-ma-gan
Past Def	bar-dɪ	bar-ma-dɪ
Fut Indef	bar-ɪr	bar-ma-s
Fut Def	bar-açak	bar-ma-yaçak

[14] with the vowel choice determined by the vowel harmony rules

Mood

Currently, KaTMorph can process verbs in the imperative and indicative moods[15]. Verbs in the imperative mood denote order, request, or invitation and conjugate by person and number for the 2nd and 3rd Person. The conjugation for the verb *kilergə* 'to come' in the imperative mood can be seen in Table 16.

Table 16. Imperative Mood Conjugation

Person/Number	Singular	Plural
1	--	--
2	kil	kil-egez
3	kil-sen	kil-sennər

Note that the translations for the 3rd person forms are equivalent to the English 'Let him/her/it/them *verb*' or 'Make her/him/it/them *verb*'. Sentence (31) exemplifies the use of the verb in the imperative mood.

(31) Бәхет килсә, бүген кил-сен (tatar folk song)
Bəxet kilsə, bügen kil-sen
bAxet kilsA, bUgen kil-sen
happiness come-Cond, today come-Imp-P3-Sg
'If happiness comes, let it come today'

Tatar verb conjugation includes adding tense or mood morphemes, followed by personal endings. There are two types of the personal endings. Type I is used after the past indefinite and future tenses; Type II is for the past definite tense. The present tense personal endings are of Type I for plural forms and of mixed type for singular. Table 17 lists the allomorphs of the personal suffixes for both types. Full verb paradigm for each tense mentioned will be presented in the corresponding sections of the Tense Morphology section.

[15] The subjunctive and conditional mood suffixes can be added in the future version of the program after more research has been completed.

Table 17. Verb Personal Suffixes

	Type I		Type II	
Singular	back	front	back	front
P1	mɪn	men	m	m
P2	sɪŋ	seŋ	ŋ	ŋ
P3	NULL	NULL	NULL	NULL
Plural				
P1	bɪz	bez	k	k
P2	sɪz	sez	gɪz	gez
P3	lar	lər	lar	lər

Tense

Tatar verbs have three tenses with the dimension of definetness: past, present, and future. The verb referring to the action happening at the moment of speaking is in the present tense, before the moment of speaking in the past tense, and after the speech event in the future tense.

When affirmative, a verb in the present tense has –a, -ɪy, or –i suffix attached to the root, followed by personal suffixes of Type I for plural and of mixed type for singular:

1. The root ending with a consonant gets suffix –a: *bar-a* ‘goes’, *kil-ə* ‘comes’. Some of the exceptions are *yөr-i* ‘walks’, *avɪr-ɪy* ‘hurts’.
2. The root ending with a back vowel gets suffix –ɪy and the root final vowel gets deleted. Observe how the surface form differs from the ungrammatical UF:

   UF: ukɪ-ɪy

   SF: uk-ɪy

   ‘reads’
3. The root ending with a front vowel gets suffix –i and the root final vowel gets deleted:

   UF: tөze-i

   SF: tөz-i

   ‘builds’

For negated verb forms in the present tense suffixes -ɪy/-i are used: *bar-m-ɪy* ‘he does not go’*, kil-m-i* ‘he does not come’, *ukɪ-m-ɪy* ‘she does not read’, *tөze-m-i* ‘she does not

build'. The present tense paradigm for both affirmative and negated forms of the verb *tap* 'find' is presented in Table 18.

Table 18. Present Tense Paradigm

	affirmative		negative	
Person	Singular	Plural	Singular	Plural
1	tab-a-m	tab-a-bɪz	tap-m-ɪy-m	tap-m-ɪy-bɪz
2	tab-a-sɪŋ	tab-a-sɪz	tap-m-ɪy-sɪŋ	tap-m-ɪy-sɪz
3	tab-a	tab-a-lar	tap-m-ɪy	tap-m-ɪy-lar

The past tense has two aspects with their own inflections: indefinite and definite. The past indefinite is used to express an event that the speakers themselves did not observe or cannot remember well; it is commonly used for describing historical facts or in narratives (Nasibullina, 2008). The past indefinite is formed by adding the underlying suffix –gan to the root, followed by personal inflections of Type I (Nasibullina, 2008). Past definite refers to the event the reality of which is without any doubt (Nasibullina, 2008) and has the underlying morpheme –de after the root, followed by personal suffixes of Type II (see Table 19 for the past tense paradigm). Sentences 31 and 32 illustrate the difference between the definite and indefinite aspects of the past tense.

(31) Ул кичә кинога бар-ган.
Ul kiçə kinoga bar-gan
Ul kiCA kinoga bar-gan
He-Nom yesterday movie-Dat go-PastIndef-P3-Sg
'Yesterday, he supposedly went to the movies.'

(32) Ул кичә кинога бар-ды.
Ul kiçə kinoga bar-dɪ
Ul kiCA kinoga bar-dI
He-Nom yesterday movie-Dat go-PastDef-P3-Sg
'Yesterday, he went to the movies.' (The speaker either witnessed this event or is absolutely certain it happened)

Table 19. Verb Conjugation for the Past Tense

	Indefinite		Definite	
Person	Sg	PL	Sg	PL
1	tap-kan-mɪn	tap-kan-bɪz	tap-tɪ-m	tap-tɪ-k
2	tap-kan-sɪŋ	tap-kan-sɪz	tap-tɪ-ŋ	tap-tɪ-gɪz
3	tap-kan	tap-kan-nar	tap-tɪ	tap-tɪ-lar

Like the past tense, the future has indefinite and definite aspects. The definite is used when the future event will certainly happen (Nasibullina, 2008). In order to form the future definite, the underlying morpheme –açak is added to the root; to form the future indefinite, the underlying suffix -ar is added to the root, followed by the personal suffixes of Type I. Table 20 lists the future tense forms of the verb 'find'.

Table 20. Verb Conjugation for the Future Tense

	Indefinite		Definite	
Person	Sg	PL	Sg	PL
1	tab-ar-mɪn	tab-ar-bɪz	tab-açak-mɪn	tab-açak-bɪz
2	tab-ar-sɪŋ	tab-ar-sɪz	tab-açak-sɪŋ	tab-açak-sɪz
3	tab-ar	tab-ar-lar	tab-açak	tab-açak-lar

To summarize, Tatar verb paradigms defined in the processor's lexicon include (structure adopted from Underhill (1986)):

1. Verb Root
2. Negation
3. Tense
4. Personal Suffixes

Finally, example 33 demonstrates a parsed verb with all possible morphemes:

(33) Бар-ма- ды- лар- мы

bar-ma- dɪ- lar- mɪ

bar-ma- dI- lar- mI

go- Neg-PastDef- P3-PL-Ques

go-not-Past-they-?

'did they not go?'

The complete verb paradigm for *tap* 'find' is presented in Appendix A.

PRONOUNS

Tatar pronouns are subdivided into personal, demonstrative, interrogative, relative, indefinite, and possessive (Nasibullina, 2008). The current version of the morphological processor can parse or generate only personal pronouns. Adding the rest of the pronouns to the system is a matter of adding their stems to the lexicon. The following information about Tatar personal pronouns is included in the lexicon.

In Tatar, there are three persons for pronouns: the 1st is to denote the speaker, the 2nd- for person being addressed, and the 3rd- for anybody else. The pronouns are devoid of gender; the third person singular pronoun *ul* is gender neutral. The pronouns have plural forms and the 2nd Plural form coincides with the 2nd Formal form. In addition, Tatar personal pronoun paradigm includes declension by six morphological cases[16]. It should be noted that there are some irregularities in pronoun morphology. First, the pronouns *min* 'I' and *sin* 'you' are non-harmonic in dative case. Second, the paradigm of the pronoun *ul* 'he/she/it' has irregular morphological structure: even though the non-nominative forms have the phonologically consistent root *an-*, this root is not the dictionary form of the 3rd person singular pronoun, which is *ul*. Table 21 represents Tatar personal pronoun paradigm.

Table 21. Personal Pronouns Paradigm

Case	P1-Sg	P2-Sg	P3-Sg	P1-PL	P2-PL	P3-PL
Nom	min 'I'	sin 'you'	ul 'he/she/it'	bez 'we'	sez 'you'	alar 'they'
Gen	minem	sineŋ	anɪŋ	bezneŋ	sezneŋ	alarnɪŋ
Dat	miŋa	siŋa	aŋa	bezgə	sezgə	alarga
Acc	mine	sine	anɪ	bezne	sezne	alarnɪ
Abl	minnən	sinnən	annan, aŋardan	bezdən	sezdən	alardan
Loc	mində	sində	anda, aŋarda	bezdə	sezdə	alarda

If a sentence consists of only a personal pronoun, to form a polar question the underlying interrogative suffix –mɪ is added at the end of the word. The sequence of pronoun morphemes is Root + Case + Interrogative. Examples (34) illustrate pronoun morphotactics.

(34) (a) Мин- ме?

[16] The interrogative pronouns 'who' and 'what' have the case feature as well, and can be added to the pronominal lexicon in the future.

Min- me?

I –Nom- Ques

Me? (‘who, me?’)

(b) Син-ең- ме?

sin- eŋ- me?

Sin- eN- me

You-Gen-Ques

‘Yours?’

(c) Алар-да- мы?

alar- da- mɪ?

alar- da- mI

They-Loc-Ques?

At them?

‘at their place?’

ADJECTIVE

The main purpose of the current morphological processor is to analyze and produce noun, personal pronoun, and verb paradigms. Adjectives were added to the program in order to demonstrate word formation in Tatar. New nouns can be derived from adjectives by adding derivational suffixes, and likewise, new adjectives can be formed by adding suffixes to nouns. The derivational suffix is appended to the noun root or stem. Table 22 demonstrates the underlying forms of the most productive suffixes for adjective formation, which are included in the lexicon (Nasibullina, 2008).

Table 22. Adjective Derivational Morphemes

stem	suffix	example
noun	lɪ	bolɪt-lɪ ‘cloudy’
noun	sɪz	təm-sez ‘tasteless’
noun	çan	eş-çən ‘hard-working’

The interrogative suffix can be added to the end of adjectives if this adjective is contrastive or is a nominal predicate. This is the only inflectional suffix that can be handled

for adjectives in the current version of the processor. Examples (35) show parses for adjectives.

(35) (a) матур

matur

matur-Adj

'beautiful'

(b) матур- мы?

matur- mɪ?

matur- mI?

matur-Adj+Ques

'(is he/she/it) beautiful?'

METHODOLOGY

For the past thirty years most work in computational morphology has been heavily dependent on finite-state methods, with the dominating approach based on finite-state transducers. Douglas Johnson (1972) was the first to demonstrate that phonological rewrite rules describe regular relations and, theoretically, can be implemented as finite-state transducers. Around 1980, Xerox researchers from Palo Alto Kaplan and Kay rediscovered this idea (Kaplan & Kay, 1994). Later, the Xerox Palo Alto Research Center developed algorithms for finite-state computing (Karttunen & Beesley, 2005). Since then, large-scale implementations of morphology for most European languages, Turkish, Arabic, Korean, and Japanese based on finite-state tools were developed at Xerox (Karttunen, 2001). Xerox finite-state tools are language independent tools that linguists can use to create finite-state transducers for various NLP tasks. These tools include lexc, a high-level language "to specify natural language lexicons" (Beesley & Karttunen, 2003, p. xv), and xfst, which provides an interface with regular expression compiler and access to the "algorithms of the Finite-State Calculus" (Beesley & Karttunen, 2003, p. 81).

A finite-state morphological analyzer and generator, which can be called a Lexical Transducer, is a two-sided network: the upper-side represents lexical strings and the lower-side represents surface strings. The lexical level string consists of the elements of the word's paradigm: root and tags representing grammatical features of the word. The surface string is

a word as it is represented in the original language (Altintas, 2001). The Lexical Transducer is bidirectional: in an analysis, the transducer relates a lower string to an upper string; in a generation the input string is applied to the upper side of the transducer, and if it is in the upper language the transducer will return the surface form(s) (Beesley & Karttunen, 2003).

The goal of the Lexical Transducer is to address the two main components of morphology: morphotactics and morphophonological alternation (Beesley & Karttunen, 2003). Morphotactic rules are encoded as finite-state networks in a finite-state lexicon. The finite-state lexicon, generally developed by a linguist, specifies roots, affixes, and morphotactics. It generates morphotactically well-formed strings (Beesley & Karttunen, 2003); the traditional linguistic name for such a form is an underlying form (UF). Phonological alternations co-occurring with morphological processes are implemented by replace rules as finite-state transducers[17]. These rules are usually designed by a linguist as well. Rule Transducers map an underlying form to the properly spelled surface string or form (SF) (Beesley & Karttunen, 2003). To sum up, the lexicon network, composed together with the replace rules, allows bidirectional mapping between the upper and lower strings.

Implementation

KaTMorph is implemented using Xerox's proprietary finite-state tools and techniques.

This system is divided into lexicon and rule modules. When lexicons and rules are composed, the resulting sublanguages are unioned together into a single lexical transducer which can process nouns, personal pronouns, adjectives, and verbs. In KaTMorph, lexicons are separate for verbs and nominals, each with its own set of replace rules which affect orthography. Lexicons and rules are based on the morphological and phonological processes discussed in the Kazan Tatar Language Section.

This processor works bidirectionally: it maps grammatical words to lexical forms, and vice versa. Figure 3 shows the architecture of such morphology system (Beesley & Karttunen, 2003).

[17] The xfst replace rules are traditionally called rewrite rules in linguistics.

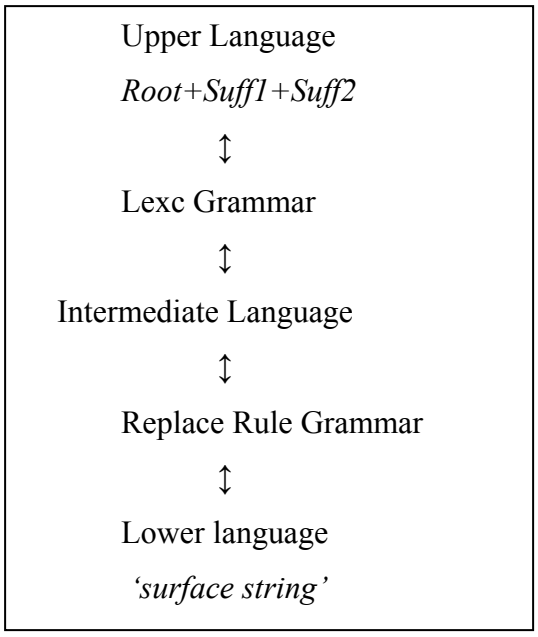

Figure 3. FST architecture.

A lexical string contains a word root with a sequence of morphemes ordered according to the morphotactic rules. These rules are explained in detail in the Tatar Morphology section. Derivational morphemes are marked by symbols such as ^DB-POS; they indicate that the root's part of speech has changed or the same part of speech with a new meaning is formed at this boundary. In order to decipher the lexical form, consult Table 7 Morphological Feature Tags on page 13 which lists the tags and their meaning used in this program. The tags for the lexical language were designed in consideration with the existing NLP systems for Turkic languages, such as the METU-Sabanci Turkish Treebank (Oflazer, Say, Hakkani-Tür, & Tür, 2003) and the Crimean Tatar morphological analyzer (Altintas & Cicekli, 2001).

Since the mapping between strings is bidirectional, it allows both production of valid words from the upper string, and analysis of a word from the lower string. When the word barmassıŋ 'you will not go' is analyzed, the returned solution has the lexical string consisting of the root and grammatical tags: bar-Verb +Neg +FutIndef +P2-Sg which reads as follows:

1. The root is 'bar'
2. Part of speech is verb
3. It is a negated form

4. Future indefinite tense
5. 2nd Person, Singular

When a word is generated from the root and a sequence of tags, such as *bar-Verb+Neg+FutIndef+P2-Sg*, each symbol is matched with the upper side symbols, and if the match is successful, the surface string *barmassɪŋ* 'you will not go' will be returned. Appendix B demonstrates the sample runs of the morphological analysis for noun paradigm *apa* 'aunt' produced by KaTMorph. The sample runs of the verb generation from the root *bar* 'go' is presented in Appendix B.

The lexicons and rules for KaTMorph were guided by Nasibullina 'Tatar language in Tables and Schemes' (2008). The choice of word roots was determined by the examples discussed by Nasibullina (2008) and by a frequency dictionary I compiled from a literary Tatar corpus[18]. Besides the root choices, the word list served as the testing data for morphological analysis during the testing and debugging phases of the development.

There are a total of 53 roots, 37 inflectional morphemes, and 8 derivational morphemes in the KaTMorph lexicon. The alphabet used in the program is Latin based with some modifications. Table 1 demonstrates the correspondence between the xfst alphabet, used in this program, and the Latin or Cyrillic alphabet used in Tatar orthography. The current version of the processor can only be executed in the environment where the xfst software has been installed[19]. The complete source code is presented in Appendix C. Let us now discuss in detail the lexicon and the replace rules implemented in KaTMorph.

Lexicon

The lexicon compiler (lexc) is a XEROX formalism used to describe finite-state networks (Beesley & Karttunen, 2003). The lexc is used to define Tatar morphotactics: word roots, suffixes, and rules for combining them. The lower-side language of the lexc transducer is an intermediate representation of a word which needs to be mapped to grammatical surface strings via the replace rules.

---

[18] I have constructed a corpus of Tatar literary language consisting of over 1,123,000 words from the publically available Tatar Electronic Library (http://kitap.net.ru). The text was randomly selected from over 20 authors dating from 1912 to the present time.

[19] link to the xfst web site**:** www.fsmbook.com

There are two separate lexicons: for verbs and for nominals. In order to keep the system "concise, maintainable and easily expandable" (Beesley & Karttunen, 2003, p. 264), the following considerations from Beesley & Karttunen (2003) were taken into account while developing lexicons:

1. To have a minimally necessary set of unique morphemes, only the underlying forms were entered. Regular morphophonological alternations to accommodate allomorphy are handled by replace rules. At the same time, there is a need to handle irregular strings in the lexc to avoid deriving them via the replace rules.
2. To design lexicons in a way that reduces the future development to the simple addition of new morphemes by a lexicographer without prior knowledge of xfst.
3. To group morphemes by grammatical features for ease of maintaining the project and better readability and accessibility.

The lexc file consists of several lexicons with unique names. The entries inside a lexicon follow these templates: 'Form Continuation Class' or 'upper:lower Continuation Class'. The Form field is a string of characters representing morphemes; 'upper:lower' format indicates a string of the upper language to be mapped to the lower; Continuation Class is the name of another lexicon within the same file to which the given form can 'continue' to form stems (Beesley & Karttunen, 2003).

The lexc baseforms, referred to as roots throughout the discussion, were chosen based on the regularities of word formation or affixation for each part of speech. The baseforms for a noun, personal pronoun, and adjective are dictionary forms; for verbs, it is the stem, which is in the imperative 2nd person singular form. These choices are dictated by the goal to keep the replace rules as simple as possible. For example, if I chose the verb baseforms to be the infinitive, dictionary form, such as *bararga* 'to go', then the infinitive marker for each verb derivation or affixation would have to be deleted. However, the inflectional and derivational suffixes attach to the verb in the imperative 2nd person singular form, hence the choice of the baseform.

Morphemes are grouped into the lexicons by the features and functions they share; morphotactic rules are implemented via continuation classes (Beesley & Karttunen, 2003). The continuation classes for nouns are derivational and inflectional suffixes; the former derive nouns and adjectives, the latter include plural, case, and personal morphemes. Verbs continuation classes include tenses, the imperative mood, personal suffixes, and negation.

Both nouns and verbs have the ability to form a yes-no question form with the Interrogative continuation class, which finalizes the word formation.

Figure 4 shows the lexicons for adjectives. The entries in the LEXICON Adj indicate A as their continuation class. The LEXICON A consists of three types of suffixes: Interog, DerivNoun2, and DerivAdj2, each has its own continuation class. The continuation class for DerivAdj2 is AdjSuff, which means that after the derivational suffix –sɪz has been added, any of the three suffixes, Interog, DerivNoun2 and DerivAdj2, can follow. The continuation class pound sign ('#') in the LEXICON Interog indicates the end of a word: this means that no morphemes can attach after the question morpheme –mɪ. Optionality of a morpheme is expressed by including an empty entry in the same LEXICON. Thus, the question morpheme is optional. Another way to express optionality is creating an intermediate LEXICON, such as the LEXICON AdjSuff in Figure 4, which makes all adjective morphemes optional (Beesley & Karttunen, 2003). The recursive word formation is expressed via loops. An example of such a loop is in the DerivAdj2 lexicon. In this segment of the lexc file, the continuation class of DerivAdj2 is AdjSuff, the LEXICON in which DerivAdj2 itself resides (Beesley & Karttunen, 2003).

Morphophonological Rules

The xfst replace rules are what linguists call phonological rewrite rules (Beesley & Karttunen, 2003). The basic conditional replace rule has the following template A -> B || L _ R where A, B, L, and R are arbitrarily complex regular expressions denoting regular languages. L and R represent the left and right context for the rule to fire. This replace rule describes a relation that maps the upper string A to the lower string B if A is between L and R (Karttunen, 1995). After the compilation of the morphotactic rules to a finite-state transducer, they are joined with the replace rules which transform the lexical forms into the surface forms.

Recall that each inflectional and derivational Tatar morpheme has multiple allomorphs conditioned by phonological processes. To keep the morphotactic rules in lexc file as general as possible, the goal is to limit the number of morphemes for each feature strictly to the underlying form. It is the replace rules that can do the work of modifying the underlying forms to the surface forms based on the phonological environment. All of the

```
LEXICON Adj
matur A ; !beautiful
iske   A ; !old
Sat   A ; !joyful

LEXICON A
-Adj:0 AdjSuff;

LEXICON AdjSuff
        Interog ;
        DerivNoun2 ;   ! optionally attach to adjective to form a noun
        DerivAdj2 ;    ! optional suffix after adjective stem to form an adjective

LEXICON DerivNoun2
+^DB-Noun:lIk N ;       !this is a recursive loop: it will take a new noun and will add
                !deriv suffixes;

LEXICON DerivAdj2
+^DB-Adj:sIz AdjSuff ;    !after adj, form adjective then a noun

LEXICON Interog
+Ques:mI  # ;
        # ;          !Optional interogatory –mI
```

Figure 4. Adjective lexicon.

underlying forms for suffixes in this processor have been arbitrarily chosen to have the [+voice, -nasal] features for the initial consonant, and the [+back] feature for vowels. Thus, the underlying form for the ablative case, for example, is –dan, and all possible allomorphs for the ablative case are: -dan, -dən, -tan,-tən, -nan, and -nən.

The replace rules in this processor are motivated by regular phonological processes existing in Tatar such as vowel harmony, assimilation, epenthesis, and deletion, and by some idiosyncrasies in word formation. They were adopted from Nasibullina (2008) and Ganiev (2000) and discussed in the Phonetic Model Section. The processes of epenthesis and deletion of segments, a vowel or a consonant, are to prevent a syllable from violating the syllable structure principles of a language[20] (Spencer, 1996). Tatar syllable structures

[20] Segment deletion and insertion are used to avoid vowel hiatus at the morpheme boundary (R.Underhill,

followed by examples are presented below (Nasibullina, 2008), where V is for Vowel, and C is for Consonant:

V: I ‘bend’

VC: ak ‘white’

CV: su ‘water’

CVC: kiŋ ‘wide’

VCC: ant ‘oath’

CVCC: kart ‘old’

The replace rules which derive grammatical word forms from lexical forms in KaTMorph are outlined in this section. These rules are presented in the order in which they apply in the derivation. It should be noted that these replace rules do not apply to word roots, with the exceptions of root-final u/ü, rule 6, and root-final voiceless consonants, rule 7. To restrict these rules from firing within a root, the suffix boundaries are marked in the intermediate strings with the symbols “+” for the morpheme boundary[21] (R. Malouf, personal communication, October 25, 2010; R. Underhill, personal communication, November 2, 2010). These rules are the generalized versions of the replace rules used in the program. For literal representations of these rules see Appendix C, Source Code in files tatar-noun-rule.regex and tatar-verb-rule.regex and see Table 23 (Karttunen, 1995) for regular expression notation.

1. Nasal Assimilation

If a noun root ends with a nasal consonant [n, ŋ, m], then –l in the plural morpheme and –d in the ablative morpheme become –n:

[l | d] -> n || [m|n|N] + _

UF: urman-lar urman-dan

SF: urman-nar urman-nan

‘forests’ ‘from forest’

2. –g deletion after N or m

---

personal communication, November 2,2010)

[21] Note, that the morpheme boundary symbol will not be seen by the users.

Table 23. Regular Expressions Notation

Symbol	Meaning
.#.	word boundary symbol
[..]	empty string
|	OR, the union operator
+	concatenation one or more times
*	Kleene-Star: concatenation zero or more times
()	optionality
[]	grouping brackets
.o.	composition
.x.	cross-product
->	replacement
_	site of replacement
||	replace rule and context separator

This rule deletes –g in the dative suffix, when preceding -N or -m is a part of the personal morpheme. However, this rule does not delete -g if the nasal consonant is a part of a root because the presence of the morphemes 'Im' or 'In' is required in the left context:

g -> [..] || + I [m | N] + _

UF: apa-ɪŋ-ga apa-ɪm-ga taŋ-ga kiem-gə

IF: apa-ɪŋ-a apa-ɪm-a taŋ-ga kiem-gə

SF[22]: apa-ŋ-a apa-m-a taŋ-ga kiem-gə

'to your aunt' 'to my aunt' 'to sunrise' 'to clothes'

3. -I deletion

This rule applies to the words marked with the personal possessive suffixes which start with the underlying vowel -ɪ. The vowel -ɪ is dropped if a noun root ends with a vowel[23], except u/ü:

+ I -> [..] || [eIiaAo] _

UF: apa-ɪm su-ɪm baş-ɪm kino-ɪm

SF: apa-m su-ɪm baş-ɪm kino-m

'my aunt' 'my water' 'my head' 'my movie'

[22] The Surface Form is derived after the application of rule 3.

[23] The vowels o/ø never occur in the final syllable in Tatar words (Nasibullina, 2008), except for the borrowings, such as the Russian borrowing *kino* 'movie'.

4.–t epenthesis with the personal morphemes

This rule inserts -t if a noun root ends with –s and an underlying personal morpheme which starts with 'ɪ' is present. It is possible that *dus* 'friend' is an irregular noun. Then its paradigm should be explicitly entered to the lexicon and not derived via morphotactic and replace rules. However, at this time I was not able to find examples of other nouns with –s final roots to check for similar behavior. This issue will be addressed in the future version of the program after more research of language data has been done.

[..] -> t || s _ + ɪ

UF: dus-ɪm

SF: dus-t-ɪm

'my friend'

5. –s deletion in the personal morphemes

If a noun root ends with a consonant or a diphthong (ending with u/ü/y), -s in the underlying 3rd Person personal morpheme is deleted:

s I -> I || [Cons | Vowel [u | ü]] + _]

UF: baş-sɪ sorau-sɪ apa-sɪ

IF[24]: baş-ɪ sorau-ɪ apa-sɪ

SF: baş-ɪ sorav-ɪ apa-sɪ

'his head' 'her question' 'his aunt'

6. u and ü become v

Root final –u or -ü become –v if it is between vowels, where the right context vowel is the beginning of an inflectional morpheme:

[u|ü] -> v || Vowel _ + I

SF: sorau-ɪm ülçəü-ɪm

UF: sorav-ɪm ülçəv-em

'my question' 'my heel'

7. Voice assimilation

[24]The IF for *sorau* is ungrammatical; the SF is derived with rule 6.

Root-final voiceless consonants become voiced in the intervocalic position, where the right context vowel is the beginning of an underlying bound morpheme. Most inflectional morphemes start with -ɪ, except the future definite and the present tense morphemes which begin with -a:

k -> g, p -> b || Vowel _ + [I | a]

UF: kolak-ɪm tap-a tap-a

SF: kolag-ɪm tab-a tab-aCak

'my ear' 'finds' 'he/she will find'

Inflectional morphemes starting with a consonant assimilate to the root final consonant voice feature. Since the underlying forms were chosen to have a voiced initial consonant, the rule devoices these consonants when preceded by a voiceless consonant.

g -> k, d -> t || VoicelessCons + _

UF: kolak-da kit-gan

SF: kolak-ta kit-kən

'to an/the ear' 'departed'

8. Vowel harmony

Due to the fact that the underlying forms of suffixes were chosen to contain a back vowel, these replace rules target the words where the root has a front vowel in the final syllable. Rule (a) maps the suffix vowels a to A and I to e, if the root final syllable has a front vowel, as in the word *keşe 'person'*. Recall that the borrowings and noun-compounds are non-harmonic, yet the rule (a) will not apply to the vowels within such roots because of the morpheme boundary restriction.

(a)

a -> A, I -> e || FrontVowel (Cons)* + (Cons) _

The rule (b) ensures that the final syllable of polysyllabic suffixes, such as –IbIz for the possessive 1st person plural, undergoes the transformation as well.

(b)

a -> A, I -> e || + (Cons) FrontVowel (Cons)+ _

UF: keşe-lar keşe-sɪz keşe-lar-ɪbɪz kitap-lar[25]

SF: keşe-lAr keşe-sez keşe-lAr-ebez kitap-lar

'people' 'without a person' 'our people' 'books'

In addition to non-harmonic borrowings, other exceptions to the vowel harmony rules are personal pronouns in dative case *siŋa* 'to you' and *miŋa* 'to me'. These exceptions are entered into the lexicon in their final surface forms. For example, the dative form of 'I' is not derived from the root with the dative suffix: the surface string *miNa* is explicitly mapped to the lexical string *min-Pron+Dat*. Because these surface strings are not derived in the dative case, there are no explicit morpheme boundaries "+", so the vowel harmony rules do not apply in these forms. However, the question forms of these words: *siŋa-mɪ* 'to you?' and *miŋa-mɪ* 'to me ?' are derived from the dative stems by the morphotactic rules. That allows the vowel in the interrogative suffix to remain [+back].

9.–y with the personal morphemes

When the personal morphemes are added to the stems which end with –y-final diphthongs, -y is deleted and -ɪ/i is converted to –e:

y + [I|i] -> e || Vowel _

UF: abɪy-ɪm ətkəy-ɪm

SF: abɪ-em ətkə-em

'my uncle' 'my father'

10. –y in negation

The present tense suffix in the negated form drops its final –y if the root final vowel is fronted. The optionality of the consonant in the left context is justified by the need to apply this rule to the words with CVV structure (*bie 'dance'*):

y -> [..] || FrontVowel (Cons)* + m + i _

UF: bar-mɪy kil-mɪy bie-mɪy kiy-mɪy

SF: bar-mɪy kil-mi bie-mi kiy-mi

'does not go' 'does not come' 'does not dance' 'does not put on'

[25] *kitap* 'book' is an Arabic borrowing

CONCLUSION AND FUTURE RESEARCH

Applications for NLP such as machine translation, information retrieval, and spell checkers may benefit from the syntactic analysis as a part of their system. Syntactic relations of agglutinative languages, like Turkic languages, are expressed by affixation; therefore, understanding morphology of such languages is crucial in constructing syntactic analysis. Finite-state methods for language processing are efficient and practical, especially for morphology of agglutinative languages because in such languages morphemes systematically encode morphosyntactic features in linear orders (Roark & Sproat, 2007).

The goal of this project is to create a tool, KaTMorph, which allows the exploration of Tatar language morphology. The tool was developed using Xerox finite-state toolkit. The main sources of descriptive generalizations used in this project are Nasibullina (2008) and Lewis (1967). The lexicon contains a finite-state description of Kazan Tatar pronominal and verb morphology, which includes morpheme classes, morphotactic and morphophonological rules.

The limitation of the rule-based morphological processors is the inability to handle words with unknown morphemes: if a root or a bound morpheme is not in the lexicon morphological analysis or production fails. This system can be expanded to handle more roots and grammatical features. To expand the roots lexicon, words from the frequency dictionary can be added to the system. A supplementary solution to this challenge is to implement a Guesser which can be used to parse or produce words with stems or morphemes even if they are not included in the lexicon (Beesley & Karttunen, 2003).

Finite-state approach for morphology is not equipped to resolve ambiguity because the context in which a word form is found is not considered in the morphological analysis. Therefore, the problem of syncretism, common for agglutinative languages[26] can not be resolved by this approach. Likewise, part of speech ambiguity, exemplified by the pair of homonyms *eş-lər* (the noun ‘work’ in the plural form) and *eşlə-r* (the verb ‘to work’ in the future indefinite tense), can not be resolved by the processor[27].

[26] For example, in Tatar, as discussed in the Noun section, the dictionary form of a noun has the functions of the nominative, indefinite accusative and indefinite genitive.

[27] The parses returned by KaTMorph for the form *eSlAr* are

The recursive nature of Tatar word formation results in recursive loops which allow an unlimited number of words and words of unlimited length. Even though recursive production is not harmful for morphological analysis, it might not be desirable for production. This is because well-formed words should occur in the language and be able to be comprehended by native speakers. To restrict such generation, we can limit the number of unique derivational morphemes a word can have by applying a 'filter' on top of the overgenerating lexicon (Beesley & Karttunen, 2003). The true challenge is determining this limit. This task can be supplemented by a Tatar corpus study and by investigating the cognitive processes involved in the usage and processing of such words. Another option is to set semantic restrictions, which would allow only certain roots to be the stems for new derivations.

To utilize the computing power of transducers, translation pairs of Tatar word stems and a language of choice, can be added in future development of this tool. Similarly, transliteration of the xfst alphabet to Cyrillic or Latin can also be implemented.

KaTMorph is a prototype of the computational morphological analyzer/generator for Kazan Tatar. In its current state it is useful for linguists who wish to understand the morphological processes of Tatar, as well as for language learners to aid in their language comprehension and the practice of word conjugation or declension . When KaTMorph contains a complete description of Tatar morphology, it will be a useful tool for large-scale NLP applications in the future.

---

eS-Noun+PL+Nom

eS-Noun+PL+Indef-Acc

eSlA-Verb+FutIndef+P3-Sg

REFERENCES

Aissen, J. (1999). Markedness and subject choice in optimality theory. *Natural Language & Linguistic Theory*, *17*(4), 673-711.

Altintas, K.(2001). *Turkish to Crimean Tatar machine translation system* (Master's thesis, Bilkent University). Retrieved from http://www.cs.bilkent.edu.tr/~ilyas/PDF/tainn2001-morph.pdf

Altintas, K., & Cicekli, I. (2001). A morphological analyzer for Crimean Tatar. Retreived from http://www.cs.bilkent.edu.tr/~ilyas/PDF/tainn2001-morph.pdf

Atalay,N., Oflazer, K., & Say, B.(2003) The annotation process in the Turkish treebank. In *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora-LINC*, Budapest, Hungary, April 13-14.

Beesley K. R., & Karttunen L. (2003). *Finite state morphology*. Stanford, CA: CSLI.

Federal State Statistics Service. (2004). All Russia population census 2002. Retrieved from http://www.perepis2002.ru/index.html?id=17

Fromkin, V., Rodman R., & Hyams, N. (2003). *An introduction to language*. Boston, MA: Heinle & Thomson.

Gadzhieva, N. Z. (1990). *Turkic languages*. Retrieved from http://www.philology.ru/linguistics4/gadzhiyeva-90.htm

Ganiev, F. A. (2000). *Tatar language: Problems and research*. Kazan, Russia: Tatar Book Publishing.

Isxakov, D. (2007). *Tatars before 2025: Demographic report*. Retrieved from http://tatpolit.ru/category/ zvezda/2007-10-05/474#comment-5688

Johnson, C. D. (1972). *Formal aspects of phonological description*. The Hague, Netherlands: Mouton Publishers.

Jurafsky, D., & Martin, J. H. (2008). *Speech and language processing: An introduction to Natural Language Processing, computational linguistics, and speech recognition*. Upper Saddle River, NJ: Pearson Education.

Kaplan, R. M., & Kay, M. (1994). *Regular models of phonological rule systems*. Retrieved from http://www.aclweb.org/anthology/J/J94/J94-3001.pdf

Karttunen, L. (1995). The replace operator. *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, Cambridge, MA, 16-23. doi:10.3115/981658.981661

Karttunen, L. (2001). Applications of finite-state transducers in Natural Language Processing. In S. Yu & A. Paun (Eds.), *Implementation and application of automata* (pp. 34-46). Heidelberg, Germany: Springer Verlag.

Karttunen, L., & Beesley, K. R. (2005). *Twenty-five years of finite-state morphology*. Retrieved from http://csli-publications.stanford.edu/koskenniemi-festschrift/8-karttunen-beesley.pdf

Lewis, G. L. (1967). *Turkish grammar*. Oxford, UK: Oxford University Press.

Nasibullina, R. N. (2008). *Tatar language in tables and schemes for Russian speaking students of elementary school*. Kazan, Russia: Gyilem Publisher.

Oflazer,K., Say, B., Hakkani-Tür, D., & Tür,G. (2003). Building a Turkish treebank. In A. Abeille (Ed.), *Building and exploiting syntactically-annotated corpora* (in press). Boston, MA: Kluwer Academic Publishers.

Republic of Tatarstan. (2010). *Tatar alphabet based on the Latin script*. Retrieved from http://www.tatar.ru/append200_a.html

Roark, B., & Sproat, R. (2007). *Computational approaches to morphology and syntax*. New York, NY: Oxford University Press.

Sabitova, I.I. (2002). *Description of word formation of Tatar lexicon*. Kazan, Russia: Fiker.

Solak, A., & Oflazer, K. (1992). Parsing agglutinative word structures and its application to spelling checking for Turkish. In *Proceedings of COLING'92—14th Conference on Computational Linguistics*, Nantes, France, July 1992. Retrieved from http://ebookpedia.net/PARSING-AGGLUTINATIVE-WORD-STRUCTURES-AND-ITS-APPLICATION-TO----.html

Spencer, A. (1996). *Phonology: Theory and description*. Oxford, UK: Blackwell Publishing.

Suleymanov, Dj. (2007). *Tatar language and infromation technology*. Kazan, Russia: Kazanskiy Federalist.

Svechnikov, A., & Sergeeva, M. (2008). *Preservation of linguistic diversity: Russian experience*. Retrieved from http://www.ifapcom.ru/files/publications/sb_eng.pdf

Tatar language. (2009).*Online Encyclopedia Krugosvet*. Retrieved from http://www.krugosvet.ru/enc/gumanitarnye_nauki/lingvistika/TATARSKI_YAZIK.html

Trost, H.(2003). Morphology. In R. Mitkov (Ed.), *The Oxford handbook of computational linguistics* (pp. 25-47). Oxford, UK: Oxford University Press.

Tzoukerman, E., Klavans, J. L., & Strzalkowski, T. (2003). Information retrieval. In R. Mitkov (Ed.), *The Oxford handbook of computational linguistics* (pp.529-544). Oxford, England: Oxford University Press.

Underhill, R. (1986). Turkish. In D. I. Slobin & K.Zimmer (Eds.), *Studies in Turkish linguistics*, (pp.7-23). Amsterdam, Netherlands: John Benjamins Publishing Co.

Zakiev, M. Z., & Ramazanova, D. B. (Eds.). (2002). *Current questions of Tatar philology*. Kazan, Russian: Fiker.

APPENDIX A

TATAR VERB PARADIGM

Table 24. Verb Paradigm of Tabarga ‘to Find’

	Imperative		Present		Past Indef		Past Def		Fututre Indef		Future Def	
	aff	neg	aff	neg	aff	neg	aff	neg	aff	neg	aff	neg
Base	tap	tap-ma	tab-a	tap-mɪy	tap-kan	tap-ma-gan	tap-tɪ	tap-ma-dɪ	tab-ar	tap-ma-s	tab-açak	tap-ma-yaçak
Singular			mix	mix	type1	type1	type2	type2	type1	mix	type1	type1
P1			tab-a-m	tap-mɪy-m	tap-kan-mɪn	tap-ma-gan-mɪn	tap-tɪ-m	tap-ma-dɪ-m	tab-ar-mɪn	tap-ma-m	tab-açak-mɪn	tap-ma-yaçak-mɪn
P2	tap	tap-ma	taba-sɪŋ	tap-mɪy-sɪŋ	tap-kan-sɪŋ	tap-ma-gan-sɪŋ	tap-tɪ-ŋ	tap-ma-dɪ-ŋ	tab-ar-sɪŋ	tap-ma-s-sɪŋ	tab-açak-sɪŋ	tap-ma-yaçak-sɪŋ
P3	tap-sɪn	tap-ma-sɪn	tab-a	tap-mɪy	tap-kan	tap-ma-gan	tap-tɪ	tap-ma-dɪ	tab-ar	tap-ma-s	tab-açak	tap-ma-yaçak
Plural			type1	type1	type1	type1	type2	type2	type1	type1	type1	type1
P1	--	--	taba-bɪz	tap-mɪy-bɪz	tap-kan-bɪz	tap-ma-gan-bɪz	tap-tɪk	tap-ma-dɪ-k	tab-ar-bɪz	tap-ma-bɪz	tab-açak-bɪz	tap-ma-yaçak-bɪz
P2	tab-ɪgɪz	tap-ma-gɪz	taba-sɪz	tap-mɪy-sɪz	tap-kan-sɪz	tap-ma-gan-sɪz	tap-tɪ-gɪz	tap-ma-dɪ-gɪz	tab-ar-sɪz	tap-ma-s-sɪz	tab-açak-sɪz	tap-ma-yaçak-sɪz
P3	tap-sɪnnar	tap-ma-sɪnnar	taba-lar	tap-mɪy-lar	tap-kan-nar	tap-ma-gan-nar	tap-tɪ-lar	tap-ma-dɪ-lar	tab-ar-lar	tap-ma-s-lar	tab-açak-lar	tap-ma-yaçak-lar

APPENDIX B

EXAMPLES OF MORPHOLOGICAL ANALYSIS AND GENERATION

Sample runs of morphological analysis of the noun inflectional paradigm *apa* 'aunt'

```
apa
apa-Noun+Sg+Indef-Acc
apa-Noun+Sg+Nom

apanI
apa-Noun+Sg+Def-Acc

apaga
apa-Noun+Sg+Dat

apadan
apa-Noun+Sg+Abl

apada
apa-Noun+Sg+Loc

apalar
apa-Noun+PL+Indef-Acc
apa-Noun+PL+Nom

apalarnI
apa-Noun+PL+Def-Acc

apalarga
apa-Noun+PL+Dat

apalardan
apa-Noun+PL+Abl

apalarda
apa-Noun+PL+Loc

apam
apa-Noun+Sg+Poss+P1+Sg+Nom


apamnI
apa-Noun+Sg+Poss+P1+Sg+Def-Acc

apama
apa-Noun+Sg+Poss+P1+Sg+Dat

apamnan
apa-Noun+Sg+Poss+P1+Sg+Abl

apamda
apa-Noun+Sg+Poss+P1+Sg+Loc

apalarIm
apa-Noun+PL+Poss+P1+Sg+Nom

apalarImnI
apa-Noun+PL+Poss+P1+Sg+Def-Acc

apalarIma
apa-Noun+PL+Poss+P1+Sg+Dat

apalarImnan
apa-Noun+PL+Poss+P1+Sg+Abl

apalarImda
apa-Noun+PL+Poss+P1+Sg+Loc

apabIz
apa-Noun+Sg+Poss+P1+PL+Nom

apabIznI
apa-Noun+Sg+Poss+P1+PL+Def-Acc

apabIzga
apa-Noun+Sg+Poss+P1+PL+Dat

apabIzdan
apa-Noun+Sg+Poss+P1+PL+Abl

apabIzda
apa-Noun+Sg+Poss+P1+PL+Loc

apalarIbIz
apa-Noun+PL+Poss+P1+PL+Nom

apalarIbIznI
apa-Noun+PL+Poss+P1+PL+Def-Acc

apalarIbIzga
apa-Noun+PL+Poss+P1+PL+Dat

apalarIbIzdan
apa-Noun+PL+Poss+P1+PL+Abl

apalarIbIzda
apa-Noun+PL+Poss+P1+PL+Loc

apaN
apa-Noun+Sg+Poss+P2+Sg+Nom

apaNnI
apa-Noun+Sg+Poss+P2+Sg+Def-Acc

apaNa
apa-Noun+Sg+Poss+P2+Sg+Dat

apaNnan
apa-Noun+Sg+Poss+P2+Sg+Abl

apaNda
apa-Noun+Sg+Poss+P2+Sg+Loc

apalarIN
apa-Noun+PL+Poss+P2+Sg+Nom

apalarINnI
apa-Noun+PL+Poss+P2+Sg+Def-Acc

apalarINa
apa-Noun+PL+Poss+P2+Sg+Dat

apalarINnan
apa-Noun+PL+Poss+P2+Sg+Abl

apalarINda
apa-Noun+PL+Poss+P2+Sg+Loc
apagIz
```

```
apa-Noun+Sg+Poss+P2+PL+Nom

apagIznI
apa-Noun+Sg+Poss+P2+PL+Def-Acc

apagIzga
apa-Noun+Sg+Poss+P2+PL+Dat

apagIzdan
apa-Noun+Sg+Poss+P2+PL+Abl

apagIzda
apa-Noun+Sg+Poss+P2+PL+Loc

apalarIgIz
apa-Noun+PL+Poss+P2+PL+Nom

apalarIgIznI
apa-Noun+PL+Poss+P2+PL+Def-Acc

apalarIgIzga
apa-Noun+PL+Poss+P2+PL+Dat

apalarIgIzdan
apa-Noun+PL+Poss+P2+PL+Abl

apalarIgIzda
apa-Noun+PL+Poss+P2+PL+Loc

apasI
apa-Noun+Sg+Poss+P3+Nom

apasIn
apa-Noun+Sg+Poss+P3+Def-Acc

apasIna
apa-Noun+Sg+Poss+P3+Dat

apasInnan
apa-Noun+Sg+Poss+P3+Abl

apasInda
apa-Noun+Sg+Poss+P3+Loc

apalarI
apa-Noun+PL+Poss+P3+Nom

apalarIn
apa-Noun+PL+Poss+P3+Def-Acc

apalarIna
apa-Noun+PL+Poss+P3+Dat

apalarInnan
apa-Noun+PL+Poss+P3+Abl

apalarInda
apa-Noun+PL+Poss+P3+Loc
```

Sample Runs of Generation of the Verb Paradigm *Bararga* 'To Go'

```
bar-Verb+Pres+P1-Sg
baram

bar-Verb+Pres+P2-Sg
barasIN

bar-Verb+Pres+P3-Sg
bara

bar-Verb+Pres+P1-PL
barabIz

bar-Verb+Pres+P2-PL
barasIz

bar-Verb+Pres+P3-PL
baralar

bar-Verb+PastDef+P1-Sg
bardIm

bar-Verb+PastDef+P2-Sg
bardIN

bar-Verb+PastDef+P3-Sg
bardI

bar+PastDef+P1-PL
bardIk

bar-Verb+PastDef+P2-PL
bardIgIz

bar-Verb+PastDef+P3-PL
bardIlar

bar-Verb+PastIndef+P1-Sg
barganmIn

bar-Verb+PastIndef+P2-Sg
bargansIN


bar-Verb+PastIndef+P3-Sg
bargan

bar-Verb+PastIndef+P1-PL
barganbIz

bar-Verb+PastIndef+P2-PL
bargansIz

bar-Verb+PastIndef+P3-PL
bargannar

bar-Verb+FutDef+P1-Sg
baraCakmIn

bar-Verb+FutDef+P2-Sg
baraCaksIN

bar-Verb+FutDef+P3-Sg
baraCak

bar-Verb+FutDef+P1-PL
baraCakbIz

bar-Verb+FutDef+P2-PL
baraCaksIz

bar-Verb+FutDef+P3-PL
baraCaklar

bar-Verb+FutIndef+P1-Sg
barIrmIn

bar-Verb+FutIndef+P2-Sg
barIrsIN

bar-Verb+FutIndef+P3-Sg
barIr

bar-Verb+FutIndef+P1-PL
barIrbIz


bar-Verb+FutIndef+P2-PL
barIrsIz

bar-Verb+FutIndef+P3-PL
barIrlar

bar-Verb+Imp-P2-Sg
bar

bar-Verb+Imp+P3-Sg
barsIn

bar-Verb+Imp+P2-PL
barIgIz

bar-Verb+Imp+P3-PL
barsInnar

bar-Verb+Pres+P1-Sg
baram


bar-Verb+Neg+FutIndef+P1-Sg
barmam

bar-Verb+Neg+PastDef+P1-Sg
barmadIm

bar-Verb+Neg+PastDef+P1-Sg+Ques
barmadImmI
```

APPENDIX C

KATMORPH XFST IMPLEMENTATION

Source Code of KaTMorph

```
#tatar-script.xfst
#xfst script that reads and compiles the lexicon and rule files
#and composes the rule network under the lexicon network
#2010, Albina Davliyeva

#Upper Language
#  |
#Lexc Grammar
#  |
#Intermediate language
#  |
# Rule Grammar
#  |
#Lower Language

clear stack

define FrontVowel [ i U O A e ] ;
define BackVowel [ I u o a ] ;
define Vowel [i U O A e I u o a ] ;

define Cons [bdgGjcClmnNrvzqpfktsSxhw] ;
define VoicedCons [bcdgGjcylmnNrvz] ;
define VoicelessCons [pfkqtsSC] ;

# read replace rules for nouns
read regex < tatar-noun-rule.regex
define Rules

# read noun lexicon
read lexc < tatar-noun-lex.txt
define LexiconN

# read replace rules for verbs
read regex < tatar-verb-rule.regex
define Rules1

# read verb lexicon
read lexc < tatar-verb-lex.txt
define LexiconV

#Compose rules
read regex [ [ LexiconN .o. Rules ] [ LexiconV .o. Rules1 ] ] ;

save stack tatar.fst
```

```
! tatar-noun-lex.txt
! This file defines noun, pronoun, adjective roots and suffixes and morphotactics
! 2010

! Do not declare suffixes as multi-character symbol, as they would be interpreted as a single character,
! and no replace rule would work

Multichar_Symbols
-Noun -Pron -Adj
+Sg +PL -Indef -Def
+Nom +Acc +Poss +Gen +Dat +Loc +Abl +Abl2
+P1 +P2 +P3
+Ques
+^DB-Noun
+^DB-Adj

LEXICON Root
        Noun ;
        Pronoun ;
        Adj ;
        Number ;
        Poss ;
        PossPL ;
        Case ;
        CasePron ;
        PossPron ;
        N ;
        A ;
        Pr ;
        Interog ;
        NSuff ;
        NSuff2 ;
        NSuff3 ;
        Deriv ;
        DerivNoun ;
        DerivNoun2 ;
        DerivAdj ;
        DerivAdj2 ;

!!!!!
! Lexical Entries by part of speech
!!!!!

!!!!!
! Nouns: noun stems are dictionary forms of nouns
!!!!!
```

```
LEXICON Noun
Ani N ; !mother
abIy N ; !uncle
alma N ; !apple
avIl N ; !village
apa N ;  !aunt, woman
atna N ; !week
baS N ;  !head
bolIt N ; !clowd
dus N ;   !friend
eS N ;    !work
kibet N ; !store
kiem N;   !clothes
kolak N ; !ear
kOzge N ; !mirror
mAktAp N ;!school
Oy N ;    !house,home
OstAl N;  !table
uku N ;   !study
UlCAU N ; !heel
UlAn N ;  !grass
urIndIk N ;!stool
uram N ;   !street
urman N ;  !forest
sorau N;   !question
sUz N ;    !word
sUzlek N ; !dictionary
taN N;     !sunset
yul N ;    !road
kitap N ;  !book (an Arabic borrowing, non-harmonic)

LEXICON Pronoun
min Pr;
sin Pr;
bez Pr;
alar Pr;
sez Pr;

LEXICON Pr
-Pron:0 CasePron ;

! Irregular forms of personal pronouns

LEXICON PossPron
min-Pron+P1+Sg+Poss:minem Interog ;
sin-Pron+P2+Sg+Poss:sineN Interog ;
ul-Pron+P3+Sg+Poss:anIN Interog ;
```

```
bez-Pron+P1+Pl+Poss:bezneN Interog ;
sez-Pron+P2+Pl+Poss:sezneN Interog ;
alar-Pron+P3+Pl+Poss:alarnIN Interog ;
min-Pron+Dat:miNa Interog ;
sin-Pron+Dat:siNa Interog ;
bez-Pron+Dat:bezgA Interog ;
sez-Pron+Dat:sezgA Interog ;
alar-Pron+Dat:alarga Interog ;
ul-Pron+Nom:ul Interog ;
ul-Pron+Acc+Def:anI Interog ;
ul-Pron+Dat:aNa Interog ;
ul-Pron+Abl:annan Interog ;
ul-Pron+Abl2:aNardan Interog ;
ul-Pron+Loc:anda Interog ;
ul-Pron+Loc2:aNarda Interog ;

LEXICON Adj
matur A ; !beautiful
iske A ;  !old
Sat A ;   !joyful

!!!
! Bound Morphemes are defined here
! The suffixes start with '7' to mark the suffix boundary to restrict the replace rules to apply only on these boundaries
!!!

!this will output part of speech tag for nouns
LEXICON N
-Noun:0 NSuff ;

!Noun root is followed by Plural or Derivational suffix
LEXICON NSuff
       Number ;
       Deriv ;

LEXICON Deriv
       DerivNoun ;
       DerivAdj ;

!suffixes to derive new nouns from noun stems
LEXICON DerivNoun
+^DB-Noun:7CI  Number ;
+^DB-Noun:7daS Number ;
+^DB-Noun:7lIk NSuff ; !recursive loop

!suffixes to form new adjectives from noun stems
LEXICON DerivAdj
+^DB-Adj:7sIz  AdjSuff ; !after noun , form adjective: dus-sIz then noun again: dus-sIz-lIk-mI
```

```
+^DB-Adj:7lI   AdjSuff ;
+^DB-Adj:7Can  AdjSuff ;
+^DB-Adj:7dagI AdjSuff ;

!number affix is followed by Possessive affix or by Null ending for Nom and Acc-Indef
LEXICON Number
+PL:7lar NSuff2 ;
+Sg:0    NSuff3 ;

LEXICON NSuff2
         PossPL ;
         Acc-Indef ; !Indef_acc case cannot follow Possessive affix

LEXICON NSuff3
           Poss ;
           Acc-Indef ; !Indef_acc case cannot followe Possessive affix

LEXICON PossCommon
+Poss+P1+Sg:7Im Case ;
+Poss+P1+PL:7IbIz Case ;
+Poss+P2+Sg:7IN Case ;
+Poss+P2+PL:7IgIz Case ;


LEXICON Poss
            PossCommon ;
+Poss+P3:7sI Case ; ! the same form for Sg or Pl possessors, but only after Sg Noun
             Case ; !Possessive is optional

LEXICON PossPL
            PossCommon;
+Poss+P3:7I  Case ; ! the same form for Sg or Pl possessors, but only after PL Noun
            Case ;  !Possessive is optional

LEXICON Case
        Nom ; !Null ending for Nominative case
        Gen ;
        Acc-Def ;
        Dat ;
        Loc ;
        Abl ;

LEXICON CasePron
        Nom ; !Null ending for Nominative case
        Acc-Def ;
        Loc ;
        Abl ;
```

```
LEXICON Nom
+Nom:0 Interog ;

LEXICON Acc-Def
+Def-Acc:7nI Interog ;

LEXICON Acc-Indef
+Indef-Acc:0 Interog ; !indefinite nouns get no case ending in accusative form

LEXICON Dat
+Dat:7ga Interog ;

LEXICON Loc
+Loc:7da Interog ;

LEXICON Abl
+Abl:7dan Interog ;

LEXICON Gen
+Gen:7nIN Interog ;

LEXICON Interog
+Ques:7mI # ; ! Interrogative suffix is the last suffix in a word
          # ; ! Optional interrogatory -mI

LEXICON A
-Adj:0 AdjSuff;

LEXICON AdjSuff
        Interog ;
        DerivNoun2 ; !this will attach to adjective to form a noun
        DerivAdj2 ;  ! after adjective stem to form adjective -without

LEXICON DerivNoun2
+^DB-Noun:7lIk NSuff ; !this is recursive loop: will take a new noun and will add deriv suffixes; change to Number to avoid the
infinite loop

LEXICON DerivAdj2
+^DB-Adj:7sIz AdjSuff ; !after adj , form adjective: Sat-sIz then noun again: dus-sIz-lIk-sIz ; also recursive loop



!tatar-verb-lex.txt
!Verb morphology is defined here.

Multichar_Symbols
-Verb +Imp+P2-Sg +Imp+P2-PL +Imp+P3-Sg +Imp+P3-PL
```

```
+Pres
+PastIndef
+PastDef
+FutDef
+FutIndef
!+FutIndef+P1-Sg
+P1-Sg +P1-PL +P2-Sg +P2-PL +P3-Sg +P3-PL
+Neg
+Ques

LEXICON Root
        Verb ;
        Imp ;
        Negat ;
        NegatFutIndef ;
        Pres ;
        Past ;
        PastPerf ;
        FutDef ;
        FutIndef ;
        FutIndefPosit ;
        PresSuffix ;
        PastSuffix ;
        FutSuffix ;
        FutNegSuffix ;
        X ;
        V ;
        Interog ;

!!!!!
! Lexical Entries
!!!!!

LEXICON Verb
bie V ;   ! dance
Ayt V ;   ! say
bar V ;   ! go
cIrla V ; ! sing
CIk V ;   ! come out
eSlA V ;  ! work
eC V ;    ! drink
kil V ;   ! come
kien V ;  ! dress
kit V ;   ! go, leave
tap V ;   ! find
tINla V ; ! listen
yarat V ; ! love
yaz V ;   ! write
```

```
ukI V ;   ! study, read
utIr V ;  ! sit

!!!!!!
! Verb Morphemes
! Morphotactic Rules for Verbs: stem + Negation + Mood + Person/Number + Question
!!!!!!

LEXICON V
-Verb:0 Negat ;

LEXICON Negat
        NegatPres ;
        NegatOther ;
         NegatFutIndef ;
        FutIndefPosit ; ! The affirmative forms for FutIndef are different from the negated

LEXICON NegatOther
+Neg:7ma X ;
         X ; ! Optional negation suffix

LEXICON NegatFutIndef
+Neg:7ma FutIndef ;!not an optional negation suffix for the future indef as the personal endings differ in affirmative and negative
forms

LEXICON NegatPres
+Neg:7m   Pres ;
          Pres ; ! Optional negation

LEXICON X
        Imp ;
        Past ;
        PastPerf ;
        FutDef ;

LEXICON Imp ! Imperative Mood Conjugates for 2nd and 3rd Person
+Imp+P2-Sg:0    # ;
+Imp+P2-PL:IgIz # ;
+Imp+P3-Sg:sIn    # ;
+Imp+P3-PL:sInnar # ;

LEXICON Pres
+Pres:7Iy PresSuffix ;

LEXICON Past
+PastIndef:7gan FutSuffix ;

LEXICON PastPerf
```

```
+PastDef:7dI PastSuffix ;

LEXICON FutIndefPosit
+FutIndef:7Ir FutSuffix ; !after l,r,t -Ir , all other -ar, replace rule ; for affirmative form

LEXICON FutIndef
+FutIndef:7s PresSuffix ; ! only after negated form

LEXICON FutDef
+FutDef:7aCak FutSuffix ;

LEXICON PresSuffix
        P1Sg ;
        P2SgPres ;
        P3Sg ;
        PersPl ;

LEXICON PastSuffix
        P1Sg;
        P2Sg ;
        P3Sg ;
        PersPlPast ;

LEXICON FutSuffix
        PersSg ;
        PersPl ;

LEXICON FutNegSuffix
       P1Sg ;
       P2SgPres ;
       P3Sg ;
       PersPl ;

LEXICON PersSg
+P1-Sg:7mIn Interog ;
+P2-Sg:7sIN Interog ;
+P3-Sg:0    Interog ;

LEXICON PersPl
+P1-PL:7bIz Interog ;
+P2-PL:7sIz Interog ;
+P3-PL:7lar Interog ;

LEXICON PersPlPast
+P1-PL:7k Interog ;
+P2-PL:7gIz Interog ;
+P3-PL:7lar Interog ;
```

```
LEXICON P1Sg
+P1-Sg:7m Interog ;

LEXICON P2SgPres
+P2-Sg:7sIN Interog ;

LEXICON P2Sg
+P2-Sg:7N Interog ;

LEXICON P3Sg
+P3-Sg:0 Interog ;

LEXICON Interog
+Ques:7mI  # ;
           # ; !optional interrogatory suffix


#tatar-noun-rule.regex
#Noun Replace Rules; they are conditioned by suffix boundaries; ‘7’ indicates a morpheme boundary

[ l -> n || [mnN] 7 _  ]
.o.
[ d -> n || [mnN] 7 _  ]
.o.
[ d -> n || n I 7 _ a n ]
.o.
[ [..] -> n || I _ 7 d a .#. ]
.o.
[ g -> n || I 7 _ a ]
.o.
[ n -> [..] || n _ 7 Vowel .#. ]
.o.
[ g -> [..] || 7 I [mN] 7 _ ]
.o.
[ I -> [..] || I 7 n _ .#. ]
.o.
[ I -> [..] || [eIiaAo] 7 _ ]
.o.
[ s I -> t || I s 7 _ ]
.o.
[ [..] -> t || s 7 _ I ]
.o.
[ s I -> I || [ Cons | [ Vowel [uU]]] 7 _ ]
.o.
[ [uU] -> v || Vowel _ 7 I ]
.o.
[ k -> g, p -> b || Vowel _ 7 I ]
.o.
```

```
[ a -> A, o -> O, I -> e || FrontVowel (y) (Cons)+ 7 (Cons) _]
              .o.
[ a -> A, o -> O, I -> e || FrontVowel (y) (Cons)+ 7 (Cons) _ , 7 (Cons) FrontVowel (Cons)+ _]
              .o.
[ a -> A, o -> O, I -> e || FrontVowel (y) (Cons)+ 7 (Cons) _ , 7 (Cons) FrontVowel (Cons)+ _]
               .o.
[ a -> A, o -> O, I -> e || 7 (Cons) FrontVowel (Cons)+ _]
.o.
[ [ O A I o a ] -> [..] || Vowel _ ~.#. ]
.o.
[ y 7 [I|i] -> e || Vowel _ ]
.o.
[ g -> k, d -> t || VoicelessCons 7 _ ]
.o.
[ 7 -> [..] ] ;


# tatar-verb-rule.regex
#Verb replace rules; ‘7’ indicates a morpheme boundary

[ p -> b , k -> g || Vowel _ 7 ]
.o.
[ [..] -> y || a _ 7 a ]
.o.
[ l -> n || n 7 _ a r ]
.o.
[ s -> [..] || m a 7 _ 7 [m | b I z] ]
.o.
[ I -> a || Cons 7 _ r ]            !FutIndefPosit
.o.
[ I -> [..] || I n _ .#. , m a 7 _ r ]
.o.
[ I -> [..] || [ e | I | a | A ] 7 _ g I z ]  !Imperative Mood
.o.
[ Vowel -> [..] || Vowel _ r ]
.o.
[ I y -> a || Cons 7 _ ]                    !Present tense suffix
.o.
[ [Vowel|y] -> [..] || _ 7 I y ]
.o.
[ I y -> i || FrontVowel (y) (Cons)* 7 _  ]  !Pres suffix
.o.
[ a -> [..] || m I y 7 _ ]                   !delete Present tense affirmative suffix
.o.
[ I -> i || FrontVowel (Cons)+ 7 m _ y ]
.o.
[ y -> [..] || FrontVowel (Cons)+ 7 m i _ ]
.o.
```

```
[ a -> A, o -> O, I -> e || FrontVowel (y) (Cons)+ 7 (Cons) _]
                .o.
[ a -> A, o -> O, I -> e || FrontVowel (y) (Cons)+ 7 (Cons) _ , 7 (Cons) FrontVowel (Cons)+ _]
                .o.
[ a -> A, o -> O, I -> e || FrontVowel (y) (Cons)+ 7 (Cons) _ , 7 (Cons) FrontVowel (Cons)+ _]
                 .o.
[ a -> A, o -> O, I -> e || 7 (Cons) FrontVowel (Cons)+ _]
.o.
[ d -> t , g -> k || VoicelessCons 7 _ ]
.o.
[ 7 -> [..] ];
```