ACTA MONGOLICA
CENTRE FOR MONGOL STUDIES
National University of Mongolia

Tom. 7 (291) 2007 045–060

Composing Algorithm for the to-Cyrillic Operations

D.Tserenpil (NUM)

The major point for creation a Modern Mongolian computational parser dealing with a Cyrillic-related text is a word form (WF), i.e. a syntactic word (or a word within a sentence), which is supposed to be mostly an inflected form (IF) - a compound unit created according to the Cyrillic orthography. A word form is defined to be text delimited by space. A word form coinciding with a database stem (DB) is considered as an uninflected form and is not analyzed in the algorithm. An inflected form presents a handling stem inflected by an inflix[1] (or inflixes), which is termed as a "processing inflix" (PI) to be differentiated from its other forms.

The primary implication for a IF is a process of its composition, defining how a IF is built-up by its components: a HS and a PI (or PI's). It is a core process and is intended basically for writing in Cyrillic Mongolian, but can be widely used for different to-Cyrillic operations, like converting and translating into Cyrillic Mongolian, learning and testing of the Cyrillic Mongolian orthography and etc. This process can be determined as an inflection and is formulated as:

$$HS + PI\ (+PI\ \ldots) \rightarrow IF$$

The secondary implication for a IF is a process of its decomposition, defining how a IF is dismissed into its components: a HS and a PI (or PI's). It is a converse process and is intended for different from-Cyrillic operations, like converting and translating from Cyrillic Mongolian, morphological analyzing in Cyrillic Mongolian and etc. This process can be determined as a recognition and is formulated as:

$$IF \rightarrow HS + PI\ (+PI\ \ldots)$$

The first implication has more sense in learning purposes, while the second one is intended mostly for research purposes. For the last years our work on parsing was

[1] The new term is created as an abbreviation of "infl(ectional suff)ix".

concentrated mostly on the decomposing algorithm dealing with from-Cyrillic analyses, particularly Cyrillic to Uighuric Mongolian[2]. The design of the algorithm was taken almost entirely from the "Mongolian Grammar"[3] featuring both Cyrillic and Uighuric scripts. The more extended report of the work was presented by an American researcher Gregory R.Ech in his Master's thesis on computational linguistics "Finite State Automata. Morphological Analysis / Generation (The Journey from Mongolian Cyrillic to Mongolian Uighurjin)"[4]. The core endeavor of the research was to produce a software program which will allow a text file written in Cyrillic Mongolian to be converted into Uighuric Mongolian Unicode text. The main method employed was to parse each WF of the Cyrillic text file to its morphemic components, identify the morphemes, then generate the appropriate Uighuric forms.

Main tools involved were a smart parsing algorithm and an extensive database. The database was produced in-house and used a simple, but efficient binary search algorithm. There are two methods of convention used. The most simple is a DB(L) lookup. This entails finding the WF in the DB and outputting the Uighuric Mongolian form associated in the DB. As the number of entries of WF's in endless however, the second method is the more preferred. It uses programming logic to parse the WF down to the rudimentary root form, obtain the root Uighuric form and append any suffixes that might be needed.

This time the composing algorithm is being proposed. The algorithm is designed to be a spell-checker for the from-Cyrillic converter. Input text is assumed to be Cyrillic Mongolian text spelled correctly. Therefore output from incorrectly spelled text may not be processed correctly.

In the composing algorithm the components are defined relating to their peculiarities. So, a HS is defined by its ending, while a PI - by its beginning. Firstly, they are defined as whether a HS is ended or a PI is started with a vowel or a consonant. The square brackets in the indexations mean that a consonant or a vowel in them is given not in a general sense, but as a particular case. Furthermore, in a more specified way, the HS is indexed by the last letters of the English alphabet in a strong sequence from the end, while the PI is indexed by the first letters of the alphabet in a strong sequence from the beginning. A plus sign (+) separates a HS part from a PI part. For example, the notation:

$$HS_{[C]} + PI_{[VV]} > ...V_WC_XV_YC_Z + V_AV_B...$$

can be read as follows: a HS ending particularly with a single consonant ($HS_{[C]}$) is joined by a PI starting particularly with a long vowel ($PI_{[VV]}$), which is specified as the last letter (since the last letter of the alphabet is "$_Z$") of the PS is a consonant (C_Z), the second letter from the end (since the second letter from the end in the alphabet is "$_Y$") in the HS is a vowel (V_Y) and

[2] The basic principles, the general operating procedures and the core algorithm for this type of a converter were introduced in the article: D.Tserenpil, G.R.Eck, "Cyrillic Mongolian to Uighuric Mongolian Converter" (in English) - 'Монгол судлал' ('Mongolian Studies' - Works in Research, the School of Mongolian Language and Culture, National University of Mongolia), 2003, t.XXI, f.16, pp.94-99

[3] D.Tserenpil, R.Kullmann, "Mongolian Grammar", Hong Kong, Jensco Ltd. 1996, XXII+447 p. Revised editions: second (Hong Kong, 2001); third (Ulaanbaatar,2005)

[4] The research under the name "Greyson" was conducted at the National University of Mongolia in Ulaanbaatar and the National University of Singapore in Singapore between 2003-2005.

etc. Similarly, the first letter (since the first letter of the alphabet is "$_A$") of the HS is a vowel (V_A), the second letter from the beginning (since the second letter from the beginning in the alphabet is "$_B$") in the HS is a vowel (V_B) and etc.

The object of the algorithm is to define the all kinds of possible combinations to determine the changes, which take place when a database stem (PS) is joined by an master inflix (PI). The following 4 types of HS + PI combinations are possible:

1.1. $HS_V + PI_V$ (HS ending with a vowel + PI starting with a vowel)
1.2. $HS_C + PI_V$ (HS ending with a consonant + PI starting with a vowel)
2.1. $HS_V + PI_C$ (HS ending with a vowel + PI starting with a consonant)
2.2. $HS_C + PI_C$ (HS ending with a consonant + PI starting with a consonant)

1. $HS + PI_V$

1.1. $HS_V + PI_V$

The vowels starting an MI can be of two types:

a) so-called "я"-type vowels
b) non-"я"-type vowels.

The first ones can be only short vowels and the latter ones can be only long vowels.

1.1.1. $HS_V + PI_{[Y]} > ...V_Z + V_A...$

The "я"-type vowels include 4 letters like "я", "е", "ё", "ю" and present actually a CV ("y+a/e/o/u") syllable in Mongolian. But they are considered artificially as vowels under the influence of the Russian Cyrillic. Only three of them: "я", "е", "ё" can be PT inflixes. The vowels ending a HS_V can be of two types:

a) short vowels including "я"-type short vowels.
b) long vowels including "я"-type long vowels.

1.1.1.1. $HS_{[V]} + PI_{[Y]} > ...V_Z + V_A...$

No HS(V) is registered in Mongolian, which ends with a basic short vowel. Therefore, "я"-type vowels as a PT inflix, have no relevant stems ending with a basic short vowel for joining. But a few HS(V)'s are found, which end with a "я"-type vowel, like "уя-", "оё-". So, in this case, the only possible composition is $HS_{[Y]} + PI_{[Y]}$. For this case, the inflix joins the stem directly. For example:

уя + я = уяя
оё + ё = оёё

1.1.1.2. $HS_{[VV]} + PI_{[Y]} > ...V_YV_Z + V_A...$

No HS(V) is registered in Mongolian, which ends with a "я"-type long vowel. Therefore, "я"-type vowels as a PT inflix have no relevant stem ending with a "я"-type long vowel to be

joined. So, in this case, the only possible composition is $HS_{[VV]} + PI_{[Y]}$. For this case, the inflix joins the stem directly. For example:

асуу + я = асууя
хөө + е = хөөе
ороо + ё = орооё
үлээ + е = үлээе

1.1.2. $HS_V + PI_{[VV]} > ...V_Z + V_AV_B...$

The non-"я"-type long vowels starting an inflix (V_AV_B) can be both doubled long vowels like "аа", "ээ", "оо", "өө", "уу", "үү" and specific long vowels in Cyrillic like "ы" and "ий" distinguishing back and front language peculiarities. The vowels ending a HS can be either a short vowel or a long vowel.

1.1.2.1. $HS_{[V]} + PI_{[VV]} > ...C_YV_Z + V_AV_B...$

Short vowels ending a stem can be of two types:

a) V_Z = non-"я"-type short vowels
b) V_Z = "я"-type short vowels

Non-"я"-type short vowels include all the basic short vowels. But "я"-type short vowels include, as mentioned above, "я", "е", "ё", "ю". But no $HS_{[V]}$ is registered where V_Z = "ю".

1.1.2.1.1. $HS_{[V]} + PI_{[VV]} > ...V_Z + V_AV_B...$ (where V_Z = non-"я"-type short vowel).

As for the algorithm, non-"я"-type basic short vowels can be of two kinds:

a) V_Z = non-"и" vowel
b) V_Z = "и" vowel

1.1.2.1.1.1. $HS_{[V]} + PI_{[VV]} > ...V_Z + V_AV_B...$ (where V_Z = non-"и" vowel).

Non-"и" vowels include the basic vowels "а", "э", "о", "ө", but the other basic vowels "у" and "ү" are not included for that they never end stems in Mongolian. In this case we deal with the process of vowel assimilation. A stem ending short vowel V_Z is assimilated to the inflix starting long vowel V_AV_B, which is shown in the inflected form as V_Z has been dropped. For example:

хулгана + ууд = хулгануу д
арга + ын = аргын
эхлэ- + ээрэй = эхлээрэй
тооно + оор = тооноор
өмнө + өөс = өмнөөс
өрнө + ийн = өрнийн

1.1.2.1.1.2. $HS_{[V]} + PI_{[VV]} > ...V_Z + V_AV_B...$ (where V_Z = "и" vowel).

The vowel "и" stands odd in the Cyrillic Mongolian for its non-assimilating peculiarities. In this case we deal with a process of uncompleted assimilation, which result the forward incorporation of the V_A into the V_Z which is "и". This process of incorporation is shown in the inflected form as V_A has been dropped. All the stems ending with "и" are supposed to be masculine and only the feminine inflix starting with "ий" (instead of starting with "ы") can be joined. For example:

архи + *аа* = *архиа*
тамхи + *ийг* = *тамхийг*
орхи- + *оод* = *орхиод*
урхи + *аар* = *урхиар*

1.1.2.1.2. $HS_{[V]} + PI_{[VV]} > ...V_Z + V_AV_B...$ (where V_Z = "я"-type short vowel).

In this case we deal with a common backward assimilation considered above, but for the reason that V_A is incorporated in "я"-type short vowel V_Z, the assimilation process for the "я"-type vowel is shown in the inflected form as V_A has been dropped. Only exceptions for this kind of dropping are the inflixes starting with "ы" and "ий" which can be understood easily. For example:

хая- + *аад* = *хаяад*
уя- + *ууд* = *уяуд*
оё- + *оорой* = *оёорой*
эе + *ээ* = *эеэ*
ая + *ыг* = *аяыг*
уе + *ийн* = *уеийн*

1.1.2.2. $HS_{[VV]} + PI_{[VV]} > ...V_YV_Z + V_AV_B...$

Long vowels ending a HS can be of the following kinds:

1. V_YV_Z is a completed long vowel, where V_Z = non-"й".
   1.1. V_YV_Z is an common long vowel (doubled), where $V_Y = V_Z$
   1.2. V_YV_Z is a "я"-type long vowel, where V_VV_Z = YV
2. V_YV_Z is an uncompleted long vowel (diphthong), where V_Z = "й".

So, there are 3 kinds of long vowels:

- common long vowels, like "аа", "ээ", "оо", "өө", "уу", "үү"
- "я"-type long vowels, like "яа", "еэ", "ёо", "юу", "юү"
- diphthongs, like "ай", "эй", "ой", "уй", "үй", "ий"

For all of these 3 cases only one common process of consonant insertion takes a place. Inserted consonants can be both "н" and "г". They are differentiated on the inflix categories:

- Inserted "н" is required for:	CS	"-ы", "-ий"	(exception list)
	CS	"-аас[4]"	(exception list)
	PL.	"-ууд[2]"	
- Inserted "г" is required for:	RF	"-аа[4]"	
	ND	"-аа[4]"	
	CS	"- аар[4]"	
	NU	"-аад[4]"	

NU	"-уул^{2}"
VD	"-аад^{4}"
PT	"-аач^{4}"
VD	"-уут^{2}"
PT	"-аарай^{4}"
PT	"-аасай^{4}"
ND	"-ууштай^{2}"

Examples are:

ширээ + ий = ширээ<u>н</u>ий
уяа + аас = уяа<u>н</u>аас
далай + ууд = далай<u>н</u>ууд
суу + аа = суу<u>г</u>аа
гоёо + оо = гоёо<u>г</u>оо
бай + аасай = бай<u>г</u>аасай

Also "x" is inserted between Gen. and RF.

1.2. $HS_C + PI_V$

As mentioned before, the inflixes in the Cyrillic Mongolian start either with a "я"-type short vowel (я/е/ё) or a common long vowel. So, we need to determine whether it is a "я"- type vowel or not.

1.2.1 $HS_C + PI_{[Y]} > ...C_Z + V_A...$ (where V_A = a "я"- type short vowel).

To deal with the PT "я"-type inflixes, we need to determine whether the stem is masculine or feminine. If the stem is masculine, only "я" and "ё" should be joined. If the stem is feminine, only the "е" inflix should be joined.

1.2.1.1. $HS_C + PI_{[Y]} > ...C_Z + V_A...$ (where DS_C is masculine, i.e. V_A = "я/ё").

If the stem is masculine, we need to determine whether the masculine stem ends in "ь".

1.2.1.1.1. $HS_C + PI_{[Y]} > ...C_Z + V_A...$ (where C_Z = non-"ь")

In this case the ъ is placed between the stem and the inflix. For example:

я = яв<u>ъ</u>я
ё = ор<u>ъ</u>ё

1.2.1.1.2. $HS_C + PI_{[Y]} > ...C_Z + V_A...$ (where C_Z is "ь")

In this case, no "ъ" is needed, and the inflix is joined directly. For example:

ярь + я = ярья
хорь + ё = хорьё

1.2.1.2. $HS_C + PI_{[Y]} > ...C_Z + V_A...$ (where HS_C is feminine, i.e. V_A = "е")

If the stem is feminine, we need to place "ь" between the stem and the inflix. For example:

өг + е = өгье
эр + е = эрье
ир + е = ирье
үлд + е = үлдье

1.2.2. $\mathbf{HS_C + PI_{[VV]} > ...C_Z + V_AV_B...}$

We need to determine whether the stem ends in "ь". If the stem ends in ь, the stem is always masculine.

1.2.2.1. $\mathbf{HS_C + PI_{[VV]} > ...C_Z + V_AV_B...}$ (where C_Z is non-"ь").

In this case, we will see further whether the stem ends in "ж/ч/ш/г".

1.2.2.1.1. $\mathbf{HS_C + PI_{[VV]} > ... C_Z + V_AV_B...}$ (where C_Z is non-"ж/ч/ш/г").

In this case, we will see further whether C_Z is followed by $C^{\emptyset}$.

1.2.2.1.1.1. $\mathbf{HS_C + PI_{[VV]} > ...C_Z + V_AV_B...}$ (where C_Z is not followed by $C^{\emptyset}$).

A further consideration depends on whether C_Z closes a first syllable or a non-first syllable. Since this case considers basically a dropping of a short vowel in a closed syllable of the type $C_XV_YC_Z$, it is important to take into an account that a short vowel in a first syllable can be never dropped.

1.2.2.1.1.1.1. $\mathbf{HS_C + PI_{[VV]} > ...C_Z + V_AV_B...}$ (where C_Z closes a first syllable)

As it is stated above, no dropping of a vowel can occur in a first syllable. Therefore, there should be shown no chances in the inflected form. For example:

нам + ын = намын
нэр + ээс = нэрээс

1.2.2.1.1.1.2. $\mathbf{HS_C + PI_{[VV]} > ...C_Z + V_AV_B...}$ (where C_Z closes a non-first syllable).

A non-first syllable in the Cyrillic Mongolian can be of two types: whether the stem ends with a single consonant (...V_YC_Z) or double (not necessarily identical) consonants (...C_YC_Z).

1.2.2.1.1.1.2.1. $\mathbf{HS_{[C]} + PI_{[VV]} > ...V_YC_Z + V_AV_B...}$

In this case, the final syllable has a structure as $C_XV_YC_Z$. A dropping of the vowel V_Y depends on whether C_X is preceded by a consonant or vowel.

1.2.2.1.1.1.2.1.1. $\mathbf{HS_{[C]} + PI_{[VV]} > ...V_WC_XV_YC_Z + V_AV_B...}$

If C_X is preceded by a vowel, then V_Y always drops. For example:

намаг + аас = намгаас

эрэг + ийн = эргийн
отор + оор = отроор
чөдөр + ийг = чөдрийг

1.2.2.1.1.1.2.1.2. $HS_{[C]} + PI_{[VV]} > ...C_WC_XV_YC_Z + V_AV_B...$

If C_X is preceded by a consonant, then V_Y may or may not drop. It depends on what kind of a consonant precedes it.

1.2.2.1.1.1.2.1.2.1. $HS_{[C]} + PI_{[VV]} > ...C_WC_XV_YC_Z + V_AV_B...$ (where $C_X = C^9$).

In this case, we have to look at the previous consonant C_W to determine whether the V_Y drops or not.

1.2.2.1.1.1.2.1.2.1.1 $HS_{[C]} + PI_{[VV]} > ...C_WC_XV_YC_Z + V_AV_B...$ (where $C_W = C^9$).

In the case, V_Y is not dropped. For example:

идсэн + ийг > идсэнийг
иддэг + ээс > иддэгээс

1.2.2.1.1.1.2.1.2.1.2 $HS_{[C]} + PI_{[VV]} > ...C_WC_XV_YC_Z + V_AV_B...$ (where $C_W = C^7$).

In the case, V_Y is dropped. For example:

явсан + ыг > явсныг
явдаг + аар > явдгаар

1.2.2.1.1.1.2.1.2.2. $HS_{[C]} + PI_{[VV]} > ... C_X V_YC_Z + V_AV_B...$ (where $C_X = C^7$).

In this case, V_Y cannot be dropped. For example:

эсрэг +ийн > эсрэгийн
самбар + аас > самбараас

1.2.2.1.1.1.2.2 $HS_{[CC]} + PI_{[VV]} > ...C_YC_Z + V_AV_B...$

In this case the final syllable has a structure as $C_WV_XC_YC_Z$ and there should be shown no changes in the inflected form. For example:

амралт + ын = амралтын
эрхлэгч + ээр = эрхлэгчээр
оромж + оос = оромжоос

1.2.2.1.1.2. $HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$ where C_Z is followed by $C^{Ø}$.

There are two phenomena in the Cyrillic Mongolian. These are so-called hidden consonants: "н" and "г" at the end of a noun stem. Originally they existed in the final position of nouns. But they have disappeared in Modern Mongolian for the certain circumstances and

accordingly are not noticed in the Cyrillic script. However, they have not disappeared completely and show up when certain inflixes for a noun stem are joined. Particularly, the showing up of the hidden "н" and "г" for certain noun inflections is shown as follows:

Inflections	Requirement	Examples
PL. "-ууд[2]"	both "н" and "г"	*мориннууд, сангууд*
Gen. "-ы", "-ий"	only "н"	*морины, усний*
Gen. "-ийн"	only "г"	*ангийн*
D.L. "-д"	only "н"	*мориндо*
Abl. "-аас[4]"	both "н" and "г"	*мориноос, сангаас*
Instr. "-аар[4]"	only "г"	*сангаар*
RF. "-аа[4]"	only "г"	*сангаа*

So, the stem ending consonants can be of two types depending on whether they are followed by a so-called hidden consonant $C^{Ø}$ or not. The hidden consonants in Mongolian are "н" and "г".

1.2.2.1.1.2.1. $\mathbf{HS_C + PI_{|VV|} > ...C_Z + V_AV_B...}$ (where C_Z is followed by $C^{Ø}$ = "н")

In this case the hidden н shows up.

1.2.2.1.1.2.2. $\mathbf{HS_C + PI_{|VV|} > ...C_Z + V_AV_B...}$ (where C_Z is followed by $C^{Ø}$ = "г")

In this case the hidden г shows up.

1.2.2.1.2. $\mathbf{HS_C + PI_{|VV|} > ...C_Z + V_AV_B...}$ (where C_Z is "ж/ч/ш/г")

This is the case where the general rule of synharmonism is ignored on masculine DS_C for the Gen. and Acc. inflixes. For example:

багш + ийн > багшийн
ач + ийг > ачийг
угж + ийг > угжийг
баг + ийн > багийн

1.2.2.2. $\mathbf{HS_C + PI_{|VV|} > ...C_Z + V_AV_B...}$ (where C_Z = "ь")

This is the case where we deal only with a masculine HS's ending with a soft consonant, since soft consonants never occur in a feminine HS's. This case causes two specific processes to be happened:

a) As in the previous case, the general rule of synharmonism is ignored on masculine HS_C for the Gen. and Acc. inflixes. Compared to the previous one, this case specify that instead of joining the mentioned feminine inflixes directly, the final "ь" has to be dropped giving no changes to the inflexes. For example:

сургууль + ийн = сургуулийн
бохь + ийг = бохийг

b) the general rule of synharmonism is not ignored on masculine PS_C for the other long vowel starting inflixes. In this case the final "ь" is turned into "и" and the first vowel of the inflix is deleted. For example:

морь + оор = мориор
амь + аа > амиа

2. $HS + PI_C$

2.1. $HS_V + PI_C > ...V_Z + C_A...$

This case works for all kinds of HS_V's and PI_C's. That means any kind of PI starting with a consonant can be joined directly to any kind of HS ending with a vowel causing no changes for both components of the composition.

2.2. $HS_C + PI_C > ...C_Z + C_A...$

The PI's starting with a consonant can be of two types:

a) PI's starting with a single consonants
b) PI's starting with a double consonants

2.2.1. $HS_C + PI_{|C|} > ...C_Z + C_AV_B...$

In this case the PI is joined directly to the HS. See the Group 1 in the "Consonant starting PI's" list. For example:

яв + сан = явсан
ол + бол = олбол
ид + мээр = идмээр

2.2.2. $HS_C + PI_{|CC|} > ...C_Z + C_AC_B...$

In this case, we need to determine whether an inserted vowel is needed between the HS_C and $PI_{[CC]}$. It depends on whether the inflix starts with C^7 or C^9.

2.2.2.1. $HS_C + PI_{|CC|} > ...C_Z + C_AC_B...$ (where $C_A = C^7$)

In this case a vowel insertion is required. See the Group 2 in the "Consonant starting MI's" list. For example:

яв- + лтай = явалтай
оч- + нгоо = очингоо
хурт- шгүй = хуртэшгүй

2.2.2.2. $HS_C + PI_{|CC|} > ...C_Z + C_AC_B...$ (where $C_A = C^9$).

In this case an insertion of a vowel may or may not needed. It depends on what kind of a consonant (C^7 or C^9) the stem ends with.

2.2.2.2.1. $HS_C + PI_{[CC]} > ...C_Z + C_AC_B...$ (where $C_Z = C^7$).

If the stem ends with a C^7, then an inserted vowel is not needed. For example:

яв + схий = явсхий
ор + цгоо = орцгоо

2.2.2.2.2. $HS_C + PI_{[CC]} > ...C_Z + C_AC_B...$ (where $C_Z = C^9$).

If the stem ends with a C^9, then an inserted vowel is needed. For example:

мат + схий = матасхий
ид + цгээ = идэцгээ

A COMPOSING ALGORITHM FOR THE DATABASE STEM INFLECTION

1.	$HS + PI_V$
1.1.	$HS_V + PI_V$
1.1.1.	$HS_V + PI_{[Y]} > ... V_Z + V_A...$
1.1.1.1.	$HS_{[V]} + PI_{[Y]} > ... C_YV_Z + V_A...$ (V_Z = "я"-type vowel)
1.1.1.2.	$HS_{[VV]} + PI_{[Y]} > ... V_YV_Z + V_A...$ (V_YV_Z = non-"я"-type vowel)
1.1.2.	$HS_V + PI_{[VV]} > ...V_Z + V_AV_B...$
1.1.2.1.	$HS_{[V]} + PI_{[VV]} > ...C_YV_Z + V_AV_B...$
1.1.2.1.1.	$HS_{[V]} + PI_{[VV]} > ...C_YV_Z + V_AV_B...$ (V_Z= non-"я"-type vowel)
1.1.2.1.1.1.	$HS_{[V]} + PI_{[VV]} > ...C_YV_Z + V_AV_B...$ (V_Z = non-"и" vowel)
1.1.2.1.1.2.	$HS_{[V]} + PI_{[VV]} > ...C_YV_Z + V_AV_B...$ (V_Z = "и" vowel)
1.1.2.1.2.	$HS_{[V]} + PI_{[VV]} > ...C_YV_Z + V_AV_B...$ (V_Z = "я"-type vowel)
1.1.2.2.	$HS_{[VV]} + PI_{[VV]} > ...V_YV_Z + V_AV_B...$
1.2.	$HS_C + PI_V$
1.2.1.	$HS_C + PI_{[Y]} > ...C_Z + V_A...$
1.2.1.1.	$HS_C + PI_{[Y]} > ...C_Z + V_A...$ (DS_C is masculine i.e.V_A = "я/ё").
1.2.1.1.1.	$HS_C + PI_{[Y]} > ...C_Z + V_A...$ (C_Z = non-"ь")
1.2.1.1.2.	$HS_C + PI_{[Y]} > ...C_Z + V_A...$ (C_Z = "ь")
1.2.1.2.	$HS_C + PI_{[Y]} > ...C_Z + V_A...$ (DS_C is feminine i.e. V_A = "е").
1.2.2.	$HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$
1.2.2.1.	$HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$ (C_Z = non-"ь")
1.2.2.1.1.	$HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$ (C_Z = non-"ж/ч/ш/г")
1.2.2.1.1.1.	$HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$ (C_Z is not followed by $C^{\varnothing}$)
1.2.2.1.1.1.1.	$HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$ (C_Z closes a first syllable)
1.2.2.1.1.1.2.	$HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$ (C_Z closes a non-first syllable)
1.2.2.1.1.1.2.1.	$HS_{[C]} + PI_{[VV]} > ...C_XV_YC_Z + V_AV_B...$
1.2.2.1.1.1.2.1.1.	$HS_{[C]} + PI_{[VV]} > ...V_WC_XV_YC_Z + V_AV_B...$
1.2.2.1.1.1.2.1.2.	$HS_{[C]} + PI_{[VV]} > ...C_WC_XV_YC_Z + V_AV_B...$
1.2.2.1.1.1.2.1.2.1.	$HS_{[C]} + PI_{[VV]} > ...C_WC_XV_YC_Z + V_AV_B...$ ($C_X = C^9$)
1.2.2.1.1.1.2.1.2.1.1.	$HS_{[C]} + PI_{[VV]} > ...C_WC_XV_YC_Z + V_AV_B...$ ($C_W = C^9$)

1.2.2.1.1.1.2.1.2.1.2.	$HS_{[C]} + PI_{[VV]} > ...C_WC_XV_YC_Z + V_AV_B...$ ($C_W = C^7$)
1.2.2.1.1.1.2.1.2.2.	$HS_{[C]} + PI_{[VV]} > ...C_WC_XV_YC_Z + V_AV_B...$ ($C_X = C^7$)
1.2.2.1.1.1.2.2.	$HS_{[CC]} + PI_{[VV]} > ...C_YC_Z + V_AV_B...$
1.2.2.1.1.2.	$HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$ (C_Z is followed by $C^{\varnothing}$)
1.2.2.1.1.2.1.	$HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$ ($C^{\varnothing}$ = "н")
1.2.2.1.1.2.2.	$HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$ ($C^{\varnothing}$ = "г")
1.2.2.1.2.	$HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$ (C_Z is "ж/ч/ш/г".
1.2.2.2.	$HS_C + PI_{[VV]} > ...C_Z + V_AV_B...$ (C_Z is "ь")
2.	$HS + PI_C$
2.1.	$HS_V + PI_C$
2.2.	$HS_C + PI_C$
2.2.1.	$HS_C + PI_{[CV]} > ...C_Z + C_AV_B...$
2.2.2.	$HS_C + PI_{[CC]} > ...C_Z + C_AC_B...$
2.2.2.1.	$HS_C + PI_{[CC]} > ...C_Z + C_AC_B...$ ($C_A = C^7$)
2.2.2.2.	$HS_C + PI_{[CC]} > ...C_Z + C_AC_B...$ ($C_A = C^9$)
2.2.2.2.1.	$HS_C + PI_{[CC]} > ...C_Z + C_AC_B...$ ($C_Z = C^7$)
2.2.2.2.2.	$HS_C + PI_{[CC]} > ...C_Z + C_AC_B...$ ($C_Z = C^9$)

1. ABBREVIATION

Abl.	-	ablative case (inflix)
Acc.		accusative case (inflix)
Act.	-	active (voice)
AD	-	adword determining inflix
Adj.	-	adjective / adjectival
Adj-b.	-	adjective-building (derivational suffix)
Adv.	-	adverb / adverbial
Advers.	-	adversative (voice)
Adw.	-	adword
AM	-	adword modifying inflix
App.	-	approximate (numerals)
AS	-	aspect inflix
BS	-	broken stem
BI	-	broken inflix
C	-	consonant
C^7	-	so-called 'vowelized consonant'. There are 7 of them in the Cyrillic as "м", "н", "г", "л", "б", "в", "р". They are required to have a vowel immediately before or after at least, if not in both sides.
C^9	-	so-called 'optional consonant'. There are 9 of them in the Cyrillic as "д", "ж", "з", "с", "т", "х", "ц", "ч", "ш". They can have a vowel immediately before or after optionally, which means sometimes they can appear without a vowel in an either side.
$C^{\varnothing}$	-	so-called 'hidden consonant', which can be either "н" or "г".
Caus.	-	causative (voice)
Cb	-	case-bound suffix

CC	-	double consonants
Co.	-	coordinating
Col.	-	collective (numerals/aspect)
Com.	-	comitative case (inflix)
Comp.	-	completed action (aspect)
Comm.	-	communal (voice)
Con.	-	conjunction
CS	-	case inflix
DB	-	database
DB(L)	-	database lexeme
DB(N)	-	nominal database lexeme
DB(Non-V)	-	non-verbal database lexeme
DB(V)	-	verbal database lexeme
Def.	-	definite (tense)
DF	-	database form
DF(X)	-	database form for X
Dim.	-	diminutive (suffix)
DL	-	dative-locative case (inflix)
DR	-	derivational suffix
FS	-	flexible stem
Fu.	-	future tense inflix
Gen.	-	genitive case (inflix)
GR	-	grammatical suffix
HF	-	handling form
HS	-	handling stem
HS(N)	-	nominal handling stem
HS(Non-V)	-	non-verbal handling stem
HS(V)	-	verbal handling stem
HS_V	-	handling stem ending with a vowel
HS_C	-	handling stem ending with a consonant
$HS_{[V]}$	-	handling stem ending with a short vowel particularly
$HS_{[VV]}$	-	handling stem ending with a long vowel particularly
$HS_{[Vi]}$	-	handling stem ending with a diphthong particularly
$HS_{[Y]}$	-	handling stem ending with a "ÿ"-type short vowel particularly
$HS_{[YV]}$	-	handling stem ending with a "ÿ"-type long vowel particularly
$HS_{[C]}$	-	handling stem ending with a single consonant particularly
$HS_{[CC]}$	-	handling stem ending with a double consonant particularly
HS(R)	-	relevant handling stem
I	-	inflix (same as inflectional suffix=IS)
IF	-	inflected form
IF(X)	-	inflected form by X
Indef.	-	indefinite (tense)
Ins.	-	instrumental case (inflix)
IS	-	inflectional suffix (same as inflix=I)
Inj.	-	interjection
L	-	lexeme

MI	-	master inflix
Mu.	-	multiplying (numerals)
N	-	noun
Nb.	-	noun-building (derivational suffix)
NDS	-	non-database stem
ND	-	noun-determining inflix
Neg.	-	negative
Non-DB	-	non-database
Non-DB(L)	-	non-database lexeme
Non-F	-	non-future (tense)
Non-HF	-	non-handling form
Non-HF(R)	-	relevant non-handling form
Non-IF	-	non-inflected form
Non-P	-	non-past (tense)
Non-RD	-	non-relevant developer
Non-RS	-	non-relational suffix
NU	-	numerals inflix
Non-V	-	non-verb
Non-VD	-	non-verb determining inflix
Or.	-	ordinal (numerals)
P1	-	first person
P2	-	second person
P3	-	third person
Pas.	-	passive (voice)
PD	-	processing deriver (derivational suffix)
PI	-	processing inflix
PI_V	-	processing inflix starting with a vowel generally
PI_C	-	processing inflix starting with a consonant generally
$PI_{[V]}$	-	processing inflix starting with a short vowel particularly
$PI_{[VV]}$	-	processing inflix starting with a long vowel particularly
$PI_{[C]}$	-	processing inflix starting with a single consonant particularly
$PI_{[CC]}$	-	processing inflix starting with a double consonant particularly
PL	-	plural inflix
Post.	-	postposition
Prep.	-	preposition
Prog.	-	progressive
Pron.	-	pronoun
Pr.	-	present tense inflix
PS	-	potential stem
PS_V	-	potential stem ending with a vowel
PS_C	-	potential stem ending with a consonant
Ps.	-	past tense inflix
Ps-neg.	-	past-negative
PT	-	person-bound terminating inflix
Q-act.	-	quick action (aspect)
R	-	root

RD - relevant developer
Rec. - recurrent (aspect)
Rep. - repetitive action (aspect)
RF - reflexive inflix
RR - relevant root
RS - relational suffix
S - stem
Sn. - sign (soft or hard)
SI - suggestible inflix
SI_V - suggestible inflix starting with a vowel
SI_C - suggestible inflix starting with a consonant
TT - tense-bound terminating inflix
Temp. - temporary action (aspect)
V - verb (used with brackets, stems...)
V - vowel (used with indexes, consonants...)
Vb. - verb-building (derivational suffix)
VC - voice inflix
VD - verb-determining inflix
VI - diphthong
VV - long vowel
X - any letter in the Cyrillic, both a vowel and a consonant. It also can be the soft and hard signs.
$X_AX_BX_CX_D$... - determination for the structure of an inflix beginning. Letters are indexed from the beginning by the first letters of the English alphabet. For example, the notation $V_AV_BC_C$ defines an inflix, where the first letter from the beginning (indexed by 'A' - the first letter in the alphabet is a vowel, the second letter from the beginning (indexed by 'B' - the second letter in the alphabet) is a vowel, the third letter from the beginning (indexed 'C' - by the third letter in the alphabet) is a consonant.
... $X_WX_XX_YX_Z$ - determination for the structure of a stem ending. Letters are indexed from the end by the last letters of the English alphabet. For example, the notation $C_WC_XV_YC_Z$ defines a stem, where the last letter (indexed by 'Z' - the first letter from the end in the alphabet) is a consonant, the second letter from the end (indexed by 'Y' - the second letter from the end in the alphabet) is a vowel, the third letter from the end (indexed 'X' - by the third letter from the end in the alphabet) is a consonant, the fourth letter from the end (indexed by 'W' - the fourth letter from the end in the alphabet) is a consonant.
Y - A "я"-type short vowel
YV - A "я"-type long vowel
ZR - zero inflix
ZD - zero deriver

READING THE ABBREVIATIONS USED

FS(X)	-	flexible stem caused by 'x' (as a phoneme)
FS(X)	-	flexible stem specialized for 'X' (as a pronoun)
F(X)	-	inflected form by the 'X' (as a suffix)
Xs	-	'X' in plural
X(N)	-	'X' as a noun
X(V)	-	'X' as a verb
X(NV)	-	'X' as non-verbal parts
X&Y	-	'X' and 'Y'
X/Y	-	'X' or 'Y'

ТОВЧЛОЛ

Кирилжин бичиг рүү хийх нийлэг алгоритм

Кирилжин бичигийн алгоритм нь кирилжин бичиг рүү хийх нийлэг алгоритм, кирилжин бичигээс хийх задлаг алгоритм гэсэн хоёр төрөл байдаг. Зохиогч өмнө нь монгол хэлний үгзүйн бүтэцийн болон кирилжин бичигийн ижил хэлбэрүүдийн алгоритмыг хэвлүүлсэн бөгөөд энэхүү илтгэлдээ кирилжин бичиг рүү хийх нийлэг алгоритмын зарчим, бүтэцийг танилцуулж байна. Кирилжин бичилбэр нь үгийн хэлбэрээс бүрдэх бөгөөд үгийн хэлбэр нь хувилах үндэс, хувилгах нөхцөл гэсэн хоёр бүрдүүлэгчээс бүрддэг. Энэхүү алгоритм нь уйгуржин зэрэг монгол бичигийн өөр тогтолцооноос кирилжин бичиг рүү хөрвүүлэг хийх, кирилжин бичигийн зөв бичих дүрэмийг алгоритмчилах, гадаад хэлнээс монгол хэл рүү автомат орчуулга хийх бичилбэрийн боловсруулалтыг хангах зэрэг хавсарга хэлшинжлэлийн олон зорилтыг хангах зорилготой юм.