

# Implementation of the Khalkha Mongolian resource grammar in GF

Nyamsuren Erdenebadrakh

3rd GF Summer School  
August 27, 2013

# Overview

Introduction

Morphophonological Rule

Morphology

Phrase Structure

Clauses

Conclusion

## Khalkha Mongolian Language

Khalkha Mongolian (=Mongolian) is an Altaic language spoken in Mongolia, China and Russian. About 8 million people in the world speak Mongolian.



Figure : Distribution of Khalkha Mongolian

# Orthography

- ▶ Script
  - ▶ Existence of the Mongolian language for over 800 years
  - ▶ Since 1946 Cyrillic an official script of Khalkha Mongolian
- ▶ Phonology
  - ▶ 47 graphemes (16 consonants, 13 vowels, 4 consonants used in foreign words, 2 softness signs, 7 long vowels, 5 diphthongs)

## Typological Characteristics

- ▶ Agglutinated Morphology
- ▶ Vowel Harmony
- ▶ The lack of a gender system
- ▶ No personal suffixes on finite verbs
- ▶ SOV-structure
- ▶ Case alternation on subjects in subordinate clauses

## Vowel Harmony

- ▶ All the vowels of a mongolian word are back (masculine) (a, o, u) or they are all front (feminine) (e, oe, ue).

## Vowel Harmony

- ▶ All the vowels of a mongolian word are back (masculine) (a, o, u) or they are all front (feminine) (e, oe, ue).
- ▶ A word cannot contain back vowels and front vowels at the same time. Exception: proper names, foreign words.

## Vowel Harmony

- ▶ All the vowels of a mongolian word are back (masculine) (a, o, u) or they are all front (feminine) (e, oe, ue).
- ▶ A word cannot contain back vowels and front vowels at the same time. Exception: proper names, foreign words.
- ▶ The vowel "i" is considered neutral and can therefore occur in both front and back voweled words.



## Vowel Harmony

- ▶ All the vowels of a mongolian word are back (masculine) (a, o, u) or they are all front (feminine) (e, oe, ue).
- ▶ A word cannot contain back vowels and front vowels at the same time. Exception: proper names, foreign words.
- ▶ The vowel "i" is considered neutral and can therefore occur in both front and back voweled words.
- ▶ The vowel of the first syllable is crucial for the voweltype of the word.

## Vowel Harmony

- ▶ All the vowels of a mongolian word are back (masculine) (a, o, u) or they are all front (feminine) (e, oe, ue).
- ▶ A word cannot contain back vowels and front vowels at the same time. Exception: proper names, foreign words.
- ▶ The vowel "i" is considered neutral and can therefore occur in both front and back voweled words.
- ▶ The vowel of the first syllable is crucial for the voweltype of the word.
- ▶ In compound words the vowel type of the last part of the word determines the vowel harmony.

## Vowel Harmony

Word endings added in Mongolian inflection (of nouns, verbs) depend on the voweltype of the stem.

```

param
  VowelType = MascA | MascO | FemOE | FemE ;

vowelType : Str -> VowelType = \stem -> case stem of {
  (_ + ("a"|"я"|"y") + ?)|(_ + ? + ("a"|"я"|"y")) => MascA ;
  (_ + ("ë"|"o") + ?)|(_ + ? + ("ë"|"o"))          => MascO ;
  (_ + "ө" + ?)|(_ + ? + "ө")                       => FemOE ;
  (_ + ("э"|"ү"|"e") + ?)|(_ + ? + ("э"|"ү"|"e")) => FemE ;
  (("A"|"Я"|"Y"|"Ю")+_) | (_+("a"|"я"|"y"|"ю")+_) => MascA ;
  (("Ë"|"O")+_) | (_+("ë"|"o")+_)                  => MascO ;
  ("Ө"+_) | (_+"ө"+_)                              => FemOE ;
  (("Э"|"Y"|"E"|"И")+_) | (_+("э"|"ү"|"e"|"и")+_) => FemE ;
  - => Predef.error (["vowelType does not apply to: "] ++
    stem)
} ;

```

## Inflectional Features of Mongolian Nouns

- ▶ 2 Number (Singular, Plural)
- ▶ 8 Cases (nominative, genitive, dative, accusative, ablative, instrumental, comitative and directional)
- + Nouns can take reflexive-possessive suffixes indicating that the marked noun is possessed by the subject of the sentence.

```
lincat
  N = {s : Number => Case => Str} ;
param
  Number = Sg | Pl ;
  Case = Nom | Gen | Dat | Acc | Abl | Inst | Com | Dir ;
```

## Mongolian Noun Definitions in GF

We distinguish 2 types of nouns:

- ▶ regular nouns  $\Rightarrow$  regN;
- ▶ nouns, which have irregular plural  $\Rightarrow$  reg2N

```
mkN = overload { mkN : Str -> Noun = regN ;
                 mkN : (_, _ : Str) -> Noun = reg2N } ;
mkLN = overload { mkLN : Str -> Noun = loanN ;
                  mkLN : (_, _ : Str) -> Noun = loan2N } ;
```

The rule of vowel drop does not apply to the foreign words, so they form a separate nominal declination class  $\Rightarrow$  loanN or loan2N.

## Mongolian Noun Definitions in GF

For the correct generation of the nominal paradigms is a Declinationtype Dcl as argument type in noun declension Function mkDecl used, which 4 different types of stem and 2 vowel type are considered.

- ▶ Variant stems caused by singular/plural and the rule of vowel dropping.
- ▶ Reason for 2 different vowel types: the vowel type of the singular stem has to be changed, if the plural suffix added begins with a vowel.

```
Dcl : Type = Str -> Str -> Str -> Str ->
          VowelType => VowelType => SubstForm => Str ;
```

For the sake of shorter description number and case are combined in the type SubstForm.

SubstForm = SF Number Case

## Mongolian Noun Definitions in GF

```

mkDecl : Bool => Dcl -> Str -> Noun =
      \\drop => \dcl -> \stem ->
let
stemDr  = case drop of {
          False => stem ;
          _     => dropUnstressedVowel stem} ;
stemPl  = plSuffix stem ;
stemPlDr = case drop of {
          False => stemPl ;
          _     => dropUnstressedVowel stemPl} ;
vts     = vowelType stem ;
vtp     = case stemPl of {"" => MascA ;
          _ + ("ҮҮД"|"yyд") => vowelType (uud2!vts) ;
          _                 => vowelType stemPl}
in
{s = (dcl stem stemDr stemPl stemPlDr) ! vts ! vtp} ;

```

**mkDecl** is used for declension of nouns and proper names, depending on the parameter `drop:Bool`. The parameter **drop** is used to distinguish between declination for ordinary nouns and declination for proper names and foreign nouns.

## Dropping of the non-initial vowels

- ▶ Basically, the vowel in the first syllable of a word is stressed. The other vowels in the stem are unstressed.
- ▶ Adding a suffix beginning with a vowel causes the final unstressed vowel between consonants to drop.

Example:

(1) суртал + ын = суртлын  
 Ideology-*N.Sg.Nom*    *Gen*  
 'of Ideology'

- ▶ The dropping function in GF is `dropUnstressedVowel` and we use only in noun declination classes.



## Function dropUnstressedVowel

```
dropUnstressedVowel : Str -> Str = \stem -> case stem of {
  _ + #doubleVowel + #consonant      => stem ;
  x@(_+#c7+ #c9) + #shortVowel + y@#c7 => (x+y) ; // for
    example, суртал (engl. Ideology)
    (...)
  -                                     => stem
} ;
```

```
Lang> i -retain mongolian/ParadigmsMon.gf
31 msec
Lang> cc dropUnstressedVowel "суртал"
"суртл"
0 msec
```

## Example of noun paradigms

airplane\_N = mkN "онгоц" ;

```
s SFSgNom   : онгоц
s SFSgGen   : онгоцны
s SFSgDat   : онгоцонд
s SFSgAcc   : онгоцыг
s SFSgAbl   : онгоцноос
s SFSgInst  : онгоцоор
s SFSgCom   : онгоцтой
s SFSgDir   : онгоц руу
s SFPlNom   : онгоцнууд
s SFPlGen   : онгоцнуудын
s SFPlDat   : онгоцнуудад
s SFPlAcc   : онгоцнуудыг
s SFPlAbl   : онгоцнуудаас
s SFPlInst  : онгоцнуудаар
s SFPlCom   : онгоцнуудтай
s SFPlDir   : онгоцнууд руу
```

boss\_N = mkN "эзэн" ;

```
s SFSgNom   : эзэн
s SFSgGen   : эзний
s SFSgDat   : эзэнд
s SFSgAcc   : эзнийг
s SFSgAbl   : эзнээс
s SFSgInst  : эзнээр
s SFSgCom   : эзэнтэй
s SFSgDir   : эзэн рүү
s SFPlNom   : эзэд
s SFPlGen   : эздийн
s SFPlDat   : эздэд
s SFPlAcc   : эздийг
s SFPlAbl   : эздээс
s SFPlInst  : эздээр
s SFPlCom   : эзэдтэй
s SFPlDir   : эзэд рүү
```

## Inflectional Features of Mongolian Verbs

- ▶ Verb forms are build by attaching Voice, Aspect and Mood suffixes to the stem, in this order.
  - ▶ The verb mood can be indicative and imperative.
  - ▶ Indicative have tenses; but there are no person and (almost) no number suffixes.
- ▶ Special forms are used for building coordination and subordination of sentences.
- ▶ Participles should be part of the adjectives and used for building relative clauses.

```
Verb = {s : VerbForm => Str ;
        vtype : VType ; vt : VowelType} ;
VerbForm = VInf Case
           | VFORM Voice Aspect VTense // indicative forms
           | VIMP Directness Imperative
           | SVDS VoiceSub Subordination
           | CVDS Anteriority // coordination
           | VPART Participle ;
```

## Mongolian Verb Definitions in GF

```

regV : Str -> Verb = \inf ->
  let
    vt                = vowelType inf ;
    stem              = stemVerb inf ;
    VoiceSuffix       = chooseVoiceSuffix stem ;
    VoiceSubSuffix    = chooseVoiceSubSuffix stem ;
    CoordinationSuffix = chooseAnterioritySuffix stem ;
    ParticipleSuffix  = chooseParticipleSuffix stem ;
    SubordinationSuffix = chooseSubordinationSuffix stem
  in {
    s = table {
      VInf c          => inf ++ infSuffixes ! c ! vt ;
      VFORM vc asp te => addSuf stem
                      (combineVAT VoiceSuffix ! vc ! asp ! te) ;
      ...
    } ;
    vtype = VAct ;
    vt = vt
  } ;

```

## Function for combine the verb suffixes

Testing showed that  $((\dots (\text{stem} + \text{suffix1}) + \text{suffix2}) \dots) + \text{suffixN}$  is much slower than  $(\text{stem} + (\text{suffix1} + \dots + \text{suffixN}))$ , because we can precompute the combination of all the suffixes for the 4 possible vowel types of stems.

*Example:*

```
combineVAT : (Voice => Suffix) ->
  Voice => Aspect => VTense => Suffix = \VoiceSuf ->
                                          \\vc,asp,te =>

  let
  AspTe = case asp of {Quick => table VowelType {vt =>
    addSufVt vt (AspectSuffix!asp!vt) (VTenseSuffix!te!FemE)
  };
          _ => table VowelType {vt =>
    addSufVt vt (AspectSuffix!asp!vt) (VTenseSuffix!te!vt)}};
  ModVT = (modifyVT VoiceSuf) ! vc
  in
  table VowelType {vt =>
    addSufVt (ModVT!vt) (VoiceSuf!vc!vt) (AspTe!(ModVT!vt))};
```

## Function addSuf

The function `addSuf` concatenate a stem perhaps extended by suffixes and a suffix varying with the vowel type by choosing the appropriate suffix variant with `addSufVt`.

```
addSuf : Str -> Suffix -> Str = \stem, suffix ->
  let
    vt = vowelType stem ;
    suf = suffix ! vt
  in addSufVt vt stem suf ;
```

`addSufVt` inserted a vowel or a softness marker between stem and suffix, if needed.

## Example of verb paradigms

Incomplete: the mongolian verb have a 170 word forms!

swim\_V = mkV "сэлэх" ;

```

s (VFORM Act Simpl VPresIndef) : сэлдэг
s (VFORM Act Simpl VPresPerf)  : сэллээ
s (VFORM Act Simpl VPastComp)  : сэлэв
s (VFORM Act Simpl VPastIndef) : сэлжээ
s (VFORM Act Simpl VPastGen)   : сэлсэн
s (VFORM Act Simpl VFut)       : сэлнэ
s (VFORM Act Quick VPresIndef) : сэлсхийдэг
s (VFORM Act Quick VPresPerf)  : сэлсхийлээ
s (VFORM Act Quick VPastComp)  : сэлсхийв
s (VFORM Act Quick VPastIndef) : сэлсхийжээ
s (VFORM Act Quick VPastGen)   : сэлсхийсэн
s (VFORM Act Quick VFut)       : сэлсхийнэ
s (VFORM Act Coll VPresIndef)  : сэлцгээдэг
s (VFORM Act Coll VPresPerf)   : сэлцгээлээ
s (VFORM Act Coll VPastComp)   : сэлцгээв
s (VFORM Act Coll VPastIndef)  : сэлцгээжээ
s (VFORM Act Coll VPastGen)    : сэлцгээсэн
s (VFORM Act Coll VFut)        : сэлцгээнэ
⋮

```

## Noun Phrases

NP in Mongolian is a record type with 4 fields:

```
NounPhrase : Type = {  
  s : Case => Str;  
  n : Number ;  
  p : Person ;  
  isPron : Bool  
} ;
```

- ▶ The field "s" is an inflection table with different forms of a noun phrase.
- ▶ The fields "n" and "p" are an agreement feature of a noun phrase which used for selecting an appropriate form of other categories
- ▶ Boolean label in the field "isPron" shows a different part of speech for a noun phrase.



## Noun Phrases

In Mongolian is noun modifiers of 2 types: pre or post.

Only the last part of a noun phrase is inflected. For example,

```
DetCN det cn = {
  s = \\c => case det.isPre of {
    True  => det.s ! Nom ++ cn.s ! Sg ! c ;
    False => cn.s ! Sg ! Nom ++ det.s ! c
  } ;
  n = det.n ;
  p = P3 ;
  isPron = False
} ;
```

## Verb Phrases

```
VerbPhrase : Type = {
  s : VPForm => {fin,aux : Str} ;
  compl : Case => Str ;
  adv : Str
} ;
```

- ▶ the field "s" means an inflection table from VPForm to a tuple of two strings. The parameter VPForm has the following constructors:

```
VPForm = VPInf Case
| VPFin ClTense Anteriority Polarity
| VPIimper Polarity Bool
| VPPass ConjForm
| VPPart ClTense Polarity
| VPSub Polarity Subordination
| VPCoord Anteriority
```

- ▶ *compl* is used for complement of a verb.
- ▶ *adv* is an adverb that can be attached to a verb to build a modified verb.

## Characteristics of mongolian clauses

- ▶ The subject is linked to the predicate.
- ▶ Main clause can exist independently of other clause and the verbal predicate always takes a tense suffix.
- ▶ Subordinate clause is dominated by the main clause, has the function of one part of the main sentence, is placed always before the main sentence.
- ▶ Depending on the subordinate clause type alternates the case marker of the subject.
- ▶ A combined sentence always consists of two or more predicates. The last predicate takes a verb in tensed form. The other verbs need a coordination suffix.
- ▶ The question is expressed with an additional particle without a change in word order.

## Clause Definition in GF

```

lin
  PredVP np vp = mkClause np.s np.n vp ;
oper
  mkClause : (Case => Str) -> Number -> VPhrase -> Clause ;
  Clause = {
    s : ClTense => Anteriority => Polarity => SType => Str
  } ;

```

```

Lang> l -treebank PredVP (UsePN john_PN) (UseV walk_V)
LangGer: Johann geht
LangMon: Джон алхдаг

```

```

Lang> l -treebank PredSCVP (EmbedS (UseCl (TTAnt TPres
ASimul) PPos (PredVP (UsePron she_Pron) (UseV go_V)))) (
UseComp (CompAP (PositA good_A)))
LangGer: dass sie geht ist gut
LangMon: тэр явдаг нь сайн байдаг

```

## Current Status of Implementation

- ▶ The grammar covers all the categories and rules of the GF abstract syntax.
- ▶ An extended lexicon with 20.000 words from crawled online newspapers is built.
- ▶ Testing of the developed grammar showed for the most part a correctness.
- ▶ The several features of the mongolian subordinate clauses must be further specified.