ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume 29, Number 4, 2026, 299–312
doi: 10.59277/ROMJIST.2026.4.02

Modeling Mongolian with Formal Language

Jonathan SANDE[1] and Tseren-Onolt ISHDORJ[2,*]

[1]School of Computer Science, Mongolia International University, Ulaanbaatar, Mongolia
[2]Graduate School, Mongolian University of Science and Technology, Ulaanbaatar, Mongolia
Email: jonathan.sande@miu.edu.mn, tseren-onolt@must.edu.mn*
* Corresponding author

Abstract This paper seeks to model the Mongolian language using formal language theory, particularly employing context-free grammar (CFG), filling a gap for sentence-level analysis that has been missing for Mongolian natural language processing. Mongolian is an agglutinative language where nouns and verbs can take many forms by concatenating suffixes to a root. This makes it challenging to create a CFG, which is usually whole-word based. A generalized CFG for simple declarative sentences covering the eight noun cases of Mongolian was constructed. Roots and suffixes are handled as separate terminals. Other context-sensitive aspects of Mongolian, like vowel harmony and spelling rules are left for post-processing. A pushdown automaton (PDA) corresponding to the CFG for Mongolian was designed to recognize whether the generated sentence structure is correct or not. This lays the foundation and creates a computational baseline for more sophisticated formal language (FL) representations of Mongolian in the future, such as mildly context-sensitive (MCS) grammars. Finally, a software platform was developed to implement the Mongolian simple declarative sentence generation and recognition by CFG and PDA, respectively. The CFG in the platform, modeled after the generalized Mongolian CFG in the paper, generates 8190 unique sentences. This is 1.7% of all possible permutations of input words in sentences up to four words long, which the PDA validated.

Key-words: Context-free grammar; formal language; Mongolian; natural language processing; pushdown automata.

1. Introduction

Mongolian is a language spoken by millions across Mongolia, China, Russia, and the diaspora around the world. However, it is a low-resource language commonly categorized as belonging to the Altaic language family with its unique agglutinative characteristics. Words are built by concatenating suffixes to the ends of nouns and verbs, where each suffix modifies the grammar and meaning. This creates a high degree of variability in word forms, which leads to challenges in natural language processing (NLP). To date, Mongolian NLP work has mainly focused either at the word level, covering important tasks such as treebank creation and part-of-speech (POS) analysis, or at the contextual level, dealing with pretrained language models. However, an area that is missing for Mongolian is a more traditional formal language approach [1] for sentence-level analysis. In this paper, which is a condensation of the first author's master's thesis at Mongolia University of Science and Technology [2], the authors seek to fill that gap by creating a context-free grammar (CFG) for Mongolian as a computational baseline. Note that CFG is one of the four main formal language grammars in the so-called Chomsky hierarchy invented by Noam Chomsky [3]. Mongolian's agglutinative nature makes creating a CFG challenging because it must handle suffixes and not just whole words. Other challenges include vowel harmony and context-dependent spelling rules. On the way to the solution, the authors looked for clues in languages related to the so-called Altaic language family. For example, Finnish, Hungarian, Turkish, and Korean share some linguistic similarities with Mongolian [4].

The CFG in this paper models simple declarative sentences covering all eight noun cases of Mongolian. Agglutination was handled by treating roots and suffixes as separate terminals. Vowel harmony and other context-related spelling rules are a separate problem and can be solved as a future post-processing step. Automata theory [5] was also considered, in particular push-down automata (PDA), to validate the generated language, whether correctly constructed or not. A CFG and a PDA are equivalent ways of describing a context-free language (CFL). The PDA can recognize sentences that are generated by the corresponding CFG. This aspect of FL has applications in structure analysis and parsing for both compilers and natural language grammar. Relevant to the study, the authors used the PDA equivalent of the CFG to validate input sentences. The results showed that 1.7% of all possible sentences up to four terminals long were valid.

CFGs are inadequate to fully represent an Altaic-type language like Mongolian, but even this overly simplistic model is missing in the academic literature for Mongolian, which has its own unique characteristics separate from other Altaic languages. While the number of grammar rules and terminals used in this study were limited, it creates a computational baseline that future research can build upon. Next steps will involve moving to a MCS model [6].

The following sections will first introduce the theoretical background of CFG and PDA, foundational natural language processing research done on Mongolian and related languages, and the major characteristics of Mongolian. The latter sections will describe the process of creating a generalized CFG for Mongolian and its associated PDA recognizer.

2. Preliminaries

2.1. Context-free grammar

A pivotal influence in the field of linguistics and computation was Noam Chomsky, who wrote about the hierarchy of grammatical forms that are used in formal languages [3]. A CFG defines hierarchical phrase structure by allowing constituents to be expanded locally, one node at a time, without reference to surrounding structure. CFGs have applications in computer programming languages. However, they have also been used to model the syntactic structure of many human languages, including English [7], French [8], Chinese [9], and many more.

2.2. Pushdown automata

For every context-free grammar that describes a context-free language, there exists a pushdown automaton (PDA) that is able to recognize whether a string of terminals belongs to that language. A PDA uses a stack data structure to traverse the parse tree of a CFG to check whether it is able to generate any particular string that belongs to that language.

2.3. Foundational work for Mongolian

2.3.1. Mongolian NLP research

CFG itself requires as its foundation an accurate understanding and definition of the grammar rules that make up the language. On the linguistic side, there is a large body of work that has been done for Mongolian. This includes systematic grammar [10], pretrained BERT models [11], and lexico-semantic resources [12]. It should be noted that the grammar rules in current works on systematic grammar have not yet been defined in a form that can be used in a context-free grammar, specifically Chomsky Normal Form [13].

Another requirement for a CFG is to have a large body of correctly labeled words with their parts of speech (POS). Work such as treebank creation [14] and POS labeling [15] has been done in this field for Mongolian, and there are at least two publicly available datasets including UniMorph [12] and POS tagging [16].

Once the grammar rules and symbols are known, sentences can be parsed. Parsing is the act of breaking a sentence into its constituent parts and applying the grammar rules to generate a parse tree. Some parsing work has been done using neural networks [17], but a CFG approach is lacking for Mongolian.

2.3.2. Formal language work for languages similar to Mongolian

Although the linguistic relationship and historical connection of the languages in the Uralic and so-called Altaic language families are debated [4], there are some languages here that share some common characteristics with Mongolian. Specifically, Finnish, Hungarian, and Turkish are all agglutinative. And like Mongolian, they have aspects of vowel harmony. Finnish uses an SVO structure, but Hungarian and Turkish use a *subject-object-verb (SOV)* structure like Mongolian.

When approaching the problem of developing a formal language-based model for Mongolian, it is helpful to borrow from work done on languages that are linguistically similar to Mongolian. Hints can be taken from these languages about the nature of parsing such a grammar. Many Finnish scholars have written about formal language, but not many papers have been published on the linguistic aspects of modeling the Finnish language itself as a CFG. However, Suomilog is an open-source CFG parser for Finnish [18]. It uses the CKY algorithm and is able to parse and generate many different morphological word forms. This paper is more focused on the sentence level, but a similar approach may be useful in the future for handling the morphological forms of Mongolian words. With regard to Hungarian, work has been done in classifying the affixes [19]. Recski describes the chunking task, building a noun phrase corpora, and building a rule-based parser and grammar [20]. With grammatical meaning contained in the word itself, agglutinative languages may also exhibit a greater level of free word order, which also affects the development of a CFG. Hungarian [19] and Turkish [21] both have a high degree of free word order. This study is limited to a fixed word order, but knowing that other linguistically similar languages have this characteristic shows the need to do further research in this area for Mongolian.

2.4. Lexical integrity

According to the Lexical Integrity Principle (LIP), "words are built out of different structural elements and by different principles of composition than syntactic phrases" [22]. The rules that govern the formation of words (morphology) are different than those that govern the order of words in sentences (syntax). Word order is flexible, but root and affix order is fixed [22].

This study relaxed the requirements of LIP by treating syntax terminals (whole words) and morphological terminals (roots and suffixes) as both in the same category. Only the expansion rules determine syntax or morphological structure. From an engineering perspective, this facilitated the creation of a set of easy to understand rules. For a small foundational study this worked well. However, the authors acknowledge the insufficiency of this approach for full language modeling of Mongolian.

3. Characteristics of Mongolian

Today, Mongolian is officially written in two different scripts. The examples in this paper will primarily be in the Cyrillic script, which is commonly used across the country of Mongolia. However, the traditional vertical script is also used for official documents in Mongolia as well as in common usage in Inner Mongolia, China.

The Mongolian Cyrillic alphabet is derived from the Russian Cyrillic alphabet and contains 35 letters. This is two more letters than Russian has. The letters "ө" and "ү" were added because Russian does not have sounds for these two Mongolian vowels. Like Japanese, Korean, and Turkish, the word order in Mongolian generally follows an SOV pattern. A literal translation of "I eat food" would be expressed as "I food eat" (Би хоол иддэг).

Beyond these general characteristics of Mongolian, it will be helpful to consider a few more special issues that are highly relevant to the development of a context-free grammar. These are agglutination, vowel harmony, and context-dependent spelling rules. They will be explained more fully below.

3.1. Agglutination

Mongolian is an agglutinative language, meaning many grammatical aspects of the language are formed by concatenating suffixes to the root word. Suffixes can primarily be divided into noun suffixes and verb suffixes.

3.1.1. Noun suffixes

Nouns have eight cases in Mongolian. The nominative case does not take a suffix, but the other seven do. The case suffixes indicate the grammatical meaning of the sentence. In English, the same meaning is usually expressed with prepositions like "to", "from", or "of".

For example, in English you might say something like "I got this from father" with "from father" being two separate words. In Mongolian, the "from" is an ablative case suffix (-аас) that is added to "father" (аав). And if you wanted to express “from *my* father”, the "my" would be added as another reflexive suffix (-аа) to the end of the word, pointing the noun ("father") back to the subject ("I").

Table 1 shows all eight cases with their suffixes and an example. Note that the directive suffix (руу), unlike the other case suffixes, is written with a space after the noun. Even so, it is still considered a suffix.

Table 1. Noun cases

Case	Suffix	Example	Meaning
Nominative	-	**Гал** унтарлаа.	The **fire** went out.
Genitive	-ын	**Галын** өнгө нь улаан.	The **fire's** color is red.
Dative	-д	**Галд** чулуу байна.	A stone is **in the fire**.
Accusative	-ыг	Тэр **галыг** унтраа.	Put out that **fire**.
Ablative	-аас	**Галаас** утаа гарсан.	Smoke came **from the fire**.
Instrumental	-аар	Хүү **галаар** тоглов.	The boy played **with fire**.
Comitative	-тай	Нүд нь **галтай**.	Her eyes were **fiery**.
Directive	руу	Тэр **гал руу** явав.	He went **toward the fire**.

3.1.2. Verb suffixes

In English, verb tenses are often expressed by a suffix. For example, walk, walks, walked, and walking. This is also true in Mongolian. However, Mongolian has many more verb suffixes than English, and they express much more than just the tense. In addition, the agglutinative nature of Mongolian allows you to append multiple suffixes to the root. Some examples of this are shown in Table 2.

Table 2. Verb with multiple suffixes

Verb	Suffix	Meaning
яв	-	go
явсан	-сан (past tense)	went
явуулсан	-уул (causative)	sent (caused to go)
явуулсангүй	-гүй (not)	didn't send
явуулчихсан	-чих (completed)	sent (finished)

Agglutinative suffixes are important to consider for developing a CFG because, unlike words, there are no spaces between the terminal symbols.

3.2. Vowel harmony

Another important characteristic of Mongolian is vowel harmony. Vowels are considered masculine (а, о, у), feminine (э, ө, ү), or neutral (и). Harmony means that Mongolian words should maintain the same gender for the vowels, though a neutral vowel could be mixed in with either. There are some exceptions to this rule, such as in foreign words, but it largely holds true. Vowel harmony also applies to case suffixes, which must match the harmony of the root noun. This is seen with the four forms of the ablative case suffix (-аас, -ээс, -оос, -өөс) in Table 3.

Table 3. Case suffix harmony

Word	Gender
аавaас	masculine (from father)
ээжээс	feminine (from mother)
номоос	masculine (from the book)
төрөөс	feminine (from the state)

Vowel harmony makes developing a CFG much more difficult because, by definition, the vowels in the suffix are context sensitive, not context-free.

3.3. Context-dependent spelling rules

In addition to the context-dependent nature of Mongolian vowel harmony, there are numerous spelling rules that are also context-based. Rather than attempt to give an exhaustive list here, they will merely be categorized as rules where letters are changed, added to, or removed from the root word when a suffix is added.

Table 4. Root word spelling mutations

Example	Spelling Rule
морь → морийг	change: ь → и
морь → мориноос	add: н
орон → орны	delete: о

In the first example of Table 4, the soft sign "ь" becomes an "и" in the presence of certain suffixes. This occurs again in the second example. However, the second example also demonstrates the addition of the "hidden (н)". Finally, the third example shows how some words drop a vowel when adding a suffix.

Such a variety of context-sensitive spelling rules makes it highly impractical to build words in Mongolian through a purely CFG approach. If the roots never changed, one could imagine handling other context-sensitive features like vowel harmony through more detailed POS labeling and rule definitions. However, mutating roots makes the problem intractable. It is better to handle the problem with post-processing or by altogether ignoring agglutination-based word building and simply using whole-word production rules.

4. A Context-free Grammar for Mongolian

With a foundation on the characteristics of Mongolian, it is now possible to begin designing a context-free grammar to model the language.

4.1. Restricting the scope

In order to make the task manageable, the paper first sets limits on what it will attempt to handle. Specifically, it only models simple declarative sentences for the eight noun cases. All noun roots are interchangeable, and post-processing for noun agglutination is out of scope. Likewise, all verbs are interchangeable and are treated as whole words, ignoring agglutination. Finally, no infinitely recursive production rules were used.

4.2. Concrete examples

Before creating a generalized CFG for Mongolian, it will be helpful to consider some concrete examples. Nouns without a case suffix are the simplest, so the authors started with that.

4.2.1. No case suffix

In Mongolian, nouns in the nominative and sometimes the accusative case take no suffix. Thus, there is no complexity related to agglutination.

Take the CFG G_{m1}

$$G_{m1} = (V, \Sigma, R, S), \tag{1}$$

where $V = \{S, N, V_{verb}\}, \Sigma = \{\text{муур}, \text{нохой}, \text{мах}, \text{ус}, \text{сүү}, \text{иддэг}, \text{уудаг}\}, R = \{S \rightarrow NV_{verb} \mid NNV_{verb}, N \rightarrow \text{муур} \mid \text{нохой} \mid \text{мах} \mid \text{ус} \mid \text{сүү}, V_{verb} \rightarrow \text{иддэг} \mid \text{уудаг}\}$. The variables ($V$) are the start symbol ($S$), a noun ($N$), and a verb ($V_{verb}$). The set of terminal symbols (Σ) includes Mongolian nouns and verbs. The production rules (R) are applied consecutively to get derivations that generate a grammatically correct Mongolian sentence. This particular grammar has 60 possible derivations.

No attempt will be made to list all of them, but here is a sample of five:

$$S \Rightarrow NV_{verb} \Rightarrow \text{нохой } V_{verb} \Rightarrow \text{нохой уудаг} \tag{2}$$
$$S \Rightarrow NV_{verb} \Rightarrow \text{муур } V_{verb} \Rightarrow \text{муур уудаг} \tag{3}$$
$$S \Rightarrow NNV_{verb} \Rightarrow \text{нохой } NV_{verb} \Rightarrow \text{нохой мах } V_{verb} \Rightarrow \text{нохой мах иддэг} \tag{4}$$
$$S \Rightarrow NNV_{verb} \Rightarrow \text{муур } NV_{verb} \Rightarrow \text{муур сүү } V_{verb} \Rightarrow \text{муур сүү уудаг} \tag{5}$$
$$S \Rightarrow NNV_{verb} \Rightarrow \text{ус } NV_{verb} \Rightarrow \text{ус нохой } V_{verb} \Rightarrow \text{ус нохой иддэг} \tag{6}$$

The meanings are (2) "dogs drink", (3) "cats drink", (4) "dogs eat meat", (5) "cats drink milk", and (6) "water eats dogs". Context-free grammars are only concerned with grammatical structure, not meaning, so the CFG generating a sentence like (6) should not be a surprise.

4.2.2. Case suffixes

Let's look at one more example to demonstrate how agglutinative word building can be done with a CFG approach. The suffixes themselves are designated as independent terminals, and the production rules indicate the conditions in which to add the suffixes. Take the CFG G_{m2}

$$G_{m2} = (V, \Sigma, R, S), \tag{7}$$

where $V = \{S, N_p, N, N_c, N_{cr}, V_{verb}\}, \Sigma = \{\text{аав}, \text{ээж}, \text{-тай}, \text{-аа}, \text{ирсэн}\}, P = \{S \rightarrow N_pV_{verb}, N_p \rightarrow NN_c \mid NN_{cr}, N_c \rightarrow N{-}\text{тай}, N_{cr} \rightarrow N_c{-}aa, N \rightarrow \text{аав} \mid \text{ээж}, V_{verb} \rightarrow \text{ирсэн}\}$. In addition to symbols defined previously, N_c is a comitative noun, and N_{cr} refers to a comitative noun with a reflexive suffix. The suffix terminals begin with a hyphen (-).

Some possible sentences that this grammar generates include the following:

- аав ээж -тай ирсэн (father came with mother)
- аав ээж -тай -аа ирсэн (father came with his mother)
- ээж аав -тай ирсэн (mother came with father)
- ээж аав -тай -аа ирсэн (mother came with her father)

The derivation for the final example in the sentence results in:

$$S \Rightarrow N_p V_{verb} \Rightarrow N N_{cr} V_{verb} \Rightarrow \text{аав } N_{cr} V_{verb} \Rightarrow \text{аав } N_c\text{-аа } V_{verb} \Rightarrow$$
$$\Rightarrow \text{аав } N\text{-тай -аа } V_{verb} \Rightarrow \text{аав ээж -тай -аа } V_{verb} \Rightarrow \text{аав ээж -тай -аа ирсэн}$$

Note that the suffixes are not properly joined to the root word. In addition to that, vowel harmony and spelling rules are not followed for the suffixes. This could be handled by a post-processing step to give "аав ээжтэйгээ ирсэн".

5. CFG Generalization for Mongolian

The examples used only a limited number of terminals. The stated restrictions in Section 4.1 were that all nouns are interchangeable and all verbs are interchangeable. That can be used to create a generalized CFG for Mongolian. Take the generic CFG G_M

$$G_M = (V, \Sigma, R, S), \tag{8}$$

where $V = \{S, N_p, N_{case}, N, S_{suf}, V_{verb}\}, \Sigma = Nouns \cup Suffixes \cup Verbs, R = \{S \to N_p V_{verb}, N_p \to N \mid NN \mid N_{case}N, N_{Case} \to NS_{suf}, N \to w$ for all $w \in Nouns, S_{suf} \to s$ for all $s \in Suffixes, V_{verb} \to v$ for all $v \in Verbs\}$. The variables V are a finite set of non-terminal symbols including the start symbol S, a noun phrase N_p, a noun with a case suffix N_{case}, a noun N, a case suffix S_{suf}, and a verb V_{verb}. Σ is a finite set of terminals made of the union of the sets $Nouns$, $Suffixes$, and $Verbs$. $Nouns$ are all base nouns in Mongolian, and $Verbs$ are all verbs in Mongolian. $Suffixes$ are the seven case suffixes in the set $\{-\text{ын}, -\text{д}, -\text{ыг}, -\text{аас}, -\text{аар}, -\text{тай}, \text{руу}\}$.

Example sentences that can be generated by G_M are shown in Table 5. These sentences are specially selected to be meaningful in Mongolian, but are still a subset of all possible sentences that this grammar generates.

Table 5. Example sentences generated by G_M

Case	Example	Meaning
Nominative	аав ирсэн	Father came.
Nominative	аав ээж ирсэн	Father and mother came.
Genitive	аав -ын ээж ирсэн	Father's mother came.
Dative	дэлгүүр -д сүү байна.	The store has milk.
Accusative	мах -ыг муур идсэн	The cat ate meat.
Ablative	гал -аас утаа гарсан	Smoke came from the fire.
Instrumental	морь -аар ээж явсан	Mother went by horse.
Comitative	багш -тай би ярьсан	I talked with a teacher.
Directive	хот руу аав явсан	Father went toward the city.

The desire to generalize the grammar as much as possible led to the choice to put the case suffix noun before the subject noun. Most of the case suffix nouns can either go before or after the subject in Mongolian. For example, the word positions for the dative case in "дэлгүүр -д сүү байна" (at the store is milk) and "сүү дэлгүүр -д байна" (milk is at the store) are both fine. However, the genitive case only allows the case suffix noun to come before the subject noun. That is, it would be wrong to say "ээж аав -ын ирсэн" (mother father's came). Thus, to make the genitive case work, the authors used the genitive case order for all of the suffixes ("аав -ын ээж ирсэн" (father's mother came) $\rightarrow N_{case}NV_{verb}$).

6. Recognizing Mongolian through Pushdown Automata

The authors wanted to know whether a string such as "аав -ын ээж ирсэн" belonged to their Mongolian CFG. Or, to put it in other terms, they wanted to know whether a genitive noun (аав -ын), followed by a noun (ээж), and ending with a verb (ирсэн) matches what the production rules of the grammar are able to generate.

As mentioned earlier, pushdown automata (PDA) use a stack data structure to traverse the parse tree of a CFG to check whether it is able to generate any particular string. This can be done in either a top-down or a bottom-up approach. A top-down traversal starts with the start symbol S and checks whether it can derive the string of terminals. A bottom-up traversal starts with the terminals and checks if it can work back to the start symbol. This paper will use the top-down approach. The following list enumerates the production rules from the generalized Mongolian CFG G_M in Section 5:

1. $S \rightarrow N_pV_{verb}$; 2. $N_p \rightarrow N$; 3. $N_p \rightarrow NN$; 4. $N_p \rightarrow N_{case}N$;
5. $N_{case} \rightarrow NS_{suf}$; 6. $N \rightarrow w$ for all $w \in Nouns$;
7. $S_{suf} \rightarrow s$ for all $s \in Suffixes$; 8. $V_{verb} \rightarrow v$ for all $v \in Verbs$.

Take the sentence "аав -ын ээж ирсэн" (father's mother came) as an example to see how a PDA recognizes a string of terminals. Table 6 shows the steps in the recognition process. The approach here and in the subsequent computation trees are modeled after Kanazawa's example and development of an English PDA [23].

The left-hand column of Table 6 shows the contents of the PDA stack, where the left-hand side of the string is the top of the stack and where the CFG's replacement rules are pushed. The middle column is the terminal string that is being checked. The right-hand column is the action taken at every step. The number in parentheses corresponds to the production rule number.

The PDA starts by pushing the start symbol S to the stack. It then continues by popping a symbol off the stack. If that symbol is non-terminal, the PDA non-deterministicly looks up a replacement rule for that symbol and pushes its replacement symbol(s) to the stack. If the popped symbol is terminal, the PDA compares it to the next terminal in the string being evaluated. If the terminals are different, then the check fails, indicating that the string does not belong to the context-free language generated by the CFG. However, if the terminals are the same, then the next symbol is popped off the stack. This continues until the stack is empty (represented by ε in Table 6). If the stack is empty and the end of the string is reached at the same time, the check passes and the string is recognized as belonging to the CFL described by the CFG.

You might wonder how the PDA predicted the correct production rule every time in Table 6. On any particular try, the wrong rule may be selected and the check would fail, but if you tried

Table 6. Top-down PDA recognizer

Stack	Terminal string	Action
S	аав -ын ээж ирсэн	start
$N_p V_{verb}$	аав -ын ээж ирсэн	predict (1)
$N_{case} N V_{verb}$	аав -ын ээж ирсэн	predict (4)
$N S_{suf} N V_{verb}$	аав -ын ээж ирсэн	predict (5)
аав $S_{suf} N V_{verb}$	аав -ын ээж ирсэн	predict (6)
$S_{suf} N V_{verb}$	-ын ээж ирсэн	match
-ын $N V_{verb}$	-ын ээж ирсэн	predict (7)
$N V_{verb}$	ээж ирсэн	match
ээж V_{verb}	ээж ирсэн	predict (6)
V_{verb}	ирсэн	match
ирсэн	ирсэн	predict (8)
ε	ε	match

every rule and at least one of them matched, then this is said to be a non-deterministic match. Implementing such an algorithm requires traversing a computation tree where each node is a CFG symbol and the children are defined by the production rules. Programmatically, this can be accomplished using another stack or recursion to facilitate backtracking.

7. Results and Discussion

7.1. Production rules

The eight production rules from the generalized Mongolian CFG G_M are implemented as described in Section 5. However, rather than attempting to include all nouns and verbs in Mongolian, the study used a small representative dataset of 13 nouns and six verbs. These nouns and verbs were chosen to increase the probability of generating a meaningful sentence. In English, the nouns and verbs were "he/she/it", "I", "Bat" (a name), "Misheel" (a name), "father", "mother", "horse", "Ulaanbaatar", "Mongolia", "America", "car", "river", "store", "came", "will come", "went", "will go", "took", and "will take". Although meaning is not a consideration in CFG generation, it is still useful to have semantically meaningful sentences when explaining CFGs to audiences unfamiliar with how they work. Terminals for the seven case suffixes of Mongolian were also included.

7.2. CFG-generated sentences

The application generates sentences with the CFG by starting with the start symbol `Symbol('S')` and repeatedly expanding it until only terminals are left. When more than one rule is available for a non-terminal input symbol (for example, `Symbol('NounPhrase')`), the application randomly selects one of them.

The following are a few sentences that the CFG generated:

- ээж авна (Mother will take [it].)
- Мишээл ээж явсан (Misheel and mother went.)

- гол би ирнэ (The river and I will come.)
- машин -тай аав явсан (Father came with a car.)
- машин -д ээж авна (Mother will take [it] in a car.)

The meaning of some of these sentences is quite questionable, but they all follow the grammatical pattern defined in the production rules. Due to nouns with case suffixes ($N_{case}N$) being only one of three choices for noun phrases in the production rules $N_p \rightarrow N \mid NN \mid N_{case}N$, only about a third of the generated results contain case suffixes.

The production rules are able to generate sentences as short as two terminals and as long as four terminals. The CFG can generate 8190 unique sentences. Increasing the number of nouns or verbs would exponentially increase the total possible sentences.

7.3. Grammar validation with PDA

The PDA feature of the application used a stack and a top-down approach to check whether user inputted terminals conform to the grammar. Backtracking was implemented with recursion. For example, the PDA accepted a sentence like "морь, -тай, Бат, ирнэ" (Bat will come with a horse) but rejected a sentence like "Мишээл, явна, Америк, руу" (Misheel to America will go).

In order to systematically validate the input strings, every permutation up to four terminals long was fed into the PDA recognizer. The results are shown in Table 7.

Table 7. Valid sentences among all permutations of terminals

Length	Permutations	Valid	Percent
1	26	0	0%
2	676	78	11.5%
3	17576	1014	5.8%
4	456976	7098	1.6%
Total	475254	8190	1.7%

As you can see, only 1.7% of the sentences were valid among the nearly half a million cumulative permutations of terminals that could form sentences up to four terminals long. This is to be expected. In natural speech, choosing four random words would be highly unlikely to produce a well-formulated grammatical sentence.

Not too much significance should be put on the number 1.7% besides that it is a small percentage. This number can vary depending on the number and type of production rules and the length of the input string. For example, in an earlier iteration of the CFG, the rules allowed strings up to five terminals long. This gave 25,137,930 permutations (possible input variations) of which only 3,840 (0.015%) were valid for the 66 rules in that CFG. However, the fact that the valid percentage is small means that there is a high percentage of invalid sentences that the PDA can flag. That's important because one area of potential future use is to use the PDA as a Mongolian grammar checker. While it can't check meaning, it can catch grammatical structures that don't conform to a recognized set of rules. The PDA is accurate to the extent that the defined production rules model the Mongolian language.

7.4. CFG and MCSG formalisms

One element of a CFG parse tree by definition knows nothing about another an element on another branch. This is fundamentally incompatible with the vowel harmony and context-dependent spelling rules that are used to agglutinatively build words in Mongolian. Ignoring vowel harmony either causes overgeneralization in the production rules, which in turn creates many grammatically incorrect terminals, or it requires a proliferation of production rules in order to pass the context down the branches of the tree. This study went with the generalized approach. The output that it produced was not grammatically correct itself but requires a further post-processing step.

Mildly context-sensitive grammars by comparison do allow for context sharing across branches of a parse tree. Whether this could handle vowel harmony and context-dependent spelling rules in Mongolian would require further research, but techniques such as tree adjoining grammars (TAG) [6] have the ability to handle grammatical hierarchies in a flexible way.

7.5. Limitations of the study

This study was an initial attempt to build a CFG and PDA for Mongolian. To that end, it used a limited number of grammar rules and vocabulary words. The only terminals used were nouns, verbs, and case suffixes. It is unclear what effect scaling the rules and terminals into the thousands would bring. Some aspects of Mongolian that were not modeled include adjectives, adverbs, and other parts of speech; recursive phrases and arbitrarily long sentences; free word and phrase order; noun root-to-suffix joining with proper spelling; verb suffixes; and exhaustive grammar rule coverage.

The study also treated noun roots and suffixes as separate terminals for the purpose of CFG sentence generation and PDA grammar recognition. For real-world applications, this would require post-processing to fix vowel harmony and context related spelling errors for CFG generation. It would also require an accurate suffix parser before PDA recognition. Neither the post-processor nor the suffix pre-processor are trivial endeavors. They were merely assumed to be perfect for this study.

8. Conclusions

This study created a context-free grammar that is able to model the agglutinative nature of the Mongolian language at a basic level. To the author's knowledge, this is the first time to publish a CFG-based analysis and model of Mongolian. The CFG models simple declarative sentences following the structure of the eight noun cases for Mongolian, seven of which contain agglutinative case suffixes. CFGs have inherent limitations with regard to handling the context-dependent aspects of a language. In Mongolian, these include vowel harmony and other context-related spelling rules. These can be handled with a post-processing step outside of the scope of CFG sentence generation. In addition to providing a Mongolian CFG, a PDA was implemented that was able to recognize valid sentences in the CFG. The application was developed using the Flutter framework and Dart programming language. This gives it the ability to run on Android, iOS, Mac, Windows, Linux and web, all without significantly changing the code base. The full source code is available on GitHub under the CC0 1.0 public domain license [24].

Initially, the authors thought that CFG and PDA might be a valid approach to building a grammar checker for Mongolian in the future, which is a tool that does not yet exist for Mongolian. However, it became clear that a CFG/PDA approach is inadequate for that and could only serve as a computational baseline for further work.

The next steps for further development involve researching how MCSG formalisms can be applied to Mongolian with its agglutinative nature, vowel harmony, and context dependent spelling rules. MCSG also holds promise for handling aspects of free word order that exist in Mongolian. The authors hope that this paper has laid a foundation for that work, not only in Mongolian, but also for other low-resource Altaic-type languages.

Acknowledgments. The authors would like to thank the "Capacity Building Project for School of Information and Communication Technology at Mongolian University of Science and Technology in Mongolia" (Contract No. P2019-00124) funded by KOICA (Korea International Cooperation Agency) for their financial support for this research project. The authors also thank anonymous reviewers for their constructive, valuable, and insightful comments that improved the quality and clarity of the manuscript.

References

[1] G. ROZENBERG and A. SALOMAA, *Handbook of Formal Languages*, Springer, Berlin, 1997.

[2] J. SANDE, *Research and Application Development for a Mongolian Context-Free Grammar*, M.Sc. thesis, Computer Science Department, Mongolian University of Science and Technology, Ulaanbaatar, Mongolia, 2025.

[3] N. CHOMSKY, *On certain formal properties of grammars*, Information and Control **2**(2), 1959, pp. 137–167.

[4] J. A. JANHUNEN, *The Unity and Diversity of Altaic*, Annual Review of Linguistics **9**, 2023, pp. 135–154.

[5] J. E. HOPCROFT and J. D. ULLMAN, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading, MA, 1979.

[6] A. K. JOSHI, K. VIJAY-SHANKER and D. WEIR, *The convergence of mildly context-sensitive grammar formalisms*, in Foundational Issues in Natural Language Processing, P. Sells, S. M. Shieber and T. Wasow, Eds., MIT Press, Cambridge, MA, 1991, pp. 31–81.

[7] C. RATHI, P. HELGE, H. DHABEKAR, K. RAJPUT and H. MUNDHRA, *A survey paper for parsing of English sentences using context-free grammar*, EasyChair Preprint, no. 7676, 2022.

[8] M. SALKOFF, *A context-free grammar of French*, Proceedings of the 8th International Conference on Computational Linguistics (COLING), Tokyo, Japan, 1980, pp. 38–45.

[9] S. MALMASI and M. DRAS, *Chinese Native Language Identification*, Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Gothenburg, Sweden, 2014, pp. 95–99.

[10] R. KULLMANN and D. TSERENPIL, *Mongolian Grammar*, 5th ed., Kullnom Verlag, Thun, 2015.

[11] E.-O. TUGULDUR, S. GUNCHIN-ISH and E. BATAA, *BERT Pretrained Models on Mongolian Datasets*. Accessed: May 16, 2026. [Online]. Available: https://github.com/tugstugi/mongolian-bert/.

[12] K. BATSUREN, A. GANBOLD, A. CHAGNAA and F. GIUNCHIGLIA, *Khalkha Mongolian (khk)*. Accessed: May 16, 2026. [Online]. Available: https://github.com/unimorph/khk.

[13] M. SIPSER, *Introduction to the Theory of Computation*, 3rd ed., Cengage Learning, Boston, MA, 2013.

[14] X. SU, G. GAO and X. YAN, *Development of Traditional Mongolian Dependency Treebank*, in Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, M. Sun et al., Eds., Springer, Berlin, Heidelberg, 2013, pp. 247–256.

[15] P. JAIMAI and O. CHIMEDDORJ, *Part of speech tagging for Mongolian corpus*, Proceedings of the 7th Workshop on Asian Language Resources (ALR7), Suntec, Singapore, 2009, pp. 103–106.

[16] GANCHIMEG, *Parts-of-speech tagging for Mongolian*. Accessed: May 16, 2026. [Online]. Available: https://github.com/ganchimegl/POS-tagger-for-Mongolian.

[17] N. LIU, X. SU, G. GAO, F. BAO and M. LU, *Morphological knowledge guided Mongolian constituent parsing*, Proceedings of International Conference on Neural Information Processing, Sydney, Australia, 2019, pp. 363–375.

[18] I. HAUHIO, *Suomilog*. Accessed: Apr. 22, 2025. [Online]. Available: https://github.com/fergusq/suomilog, 2025.

[19] E. FARKAS, C. H. A. KOSTER, P. KÖVES and M. NASZÓDI, *Towards an affix grammar for the Hungarian language*, Proceedings of Conference on Intelligent Systems, Veszprém, Hungary, 1991, pp. 223–236.

[20] G. RECSKI, *NP chunking in Hungarian*, M.A. thesis, Eötvös Loránd University of Sciences, Budapest, Hungary, 2010.

[21] İ. DÖNMEZ and E. ADALI, *Contex free grammer for Turkish*, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi **22**(2), 2018, pp. 552–561.

[22] J. BRESNAN and S. A. MCHOMBO, *The lexical integrity principle: Evidence from Bantu*, Natural Language & Linguistic Theory **13**(2), 1995, pp. 181–254.

[23] M. KANAZAWA, *Lecture 2*, in Formal Grammar: An Introduction, Unpublished lecture notes. Accessed: Apr. 23, 2025. [Online]. Available: https://makotokanazawa.ws.hosei.ac.jp/FormalGrammar/lecture2.pdf.

[24] J. SANDE, *Mongolian CFG/PDA*. Accessed: Nov. 20, 2025. [Online]. Available: https://github.com/suragch/mongol_cfg.