Formal Grammars and the Philippine Languages

Zhean Ganituen **Stephen Borja** **Justin Ching** **Nathaniel Oco**

November 7, 2025

Correspondence: zhean_robby_ganituen@dlsu.edu.ph
De La Salle University

Introduction

About Me

I am **Zhean** (pronunciation: “go wild with it”).

Inside the University: **MangoCats**; working on drug name (look-alike and sound-alike) similarity for the Philippine context; Filipino orthography, phonetics, and G2P; syllabic phonetic mapping and spelling nativization algorithm. [1]

Outside the University: Programming Language Theory, Parallel Algorithms, and (along, long time ago…) Ray Tracing. Criticizing web developers.

Unlike most of y’all, I **do not** want to be a software engineer or a tech entrepreneur!

[1] With Stephen, Erin, Gideon, and advisor Nathaniel Oco. Visit: `https://github.com/Mango-Cats`

"I want to has good grammars for wroted messages"

Why Formal Grammars?

“Just use ChatGPT!”

Why bother with formal grammars? Sounds like a snooze fest…

Just wrap ChatGPT in TypeScript (or whatever people do these days) like everyone else!

Transparent Formalism + Language Complexity

By constructing formal grammars for Philippine languages:

1. We build **human-interpretable** tools for Filipino digital citizens
    - Improving productivity
    - Promoting language inclusion

2. We gain insights into the **computational complexity** of human languages

Formal Languages

Formal Languages + Chomsky

Formal Language $\mathcal{L}$ over alphabet Σ:

- Set of all **valid** strings from Σ [2]

Chomsky Hierarchy classifies languages into four levels:

- **Regular** — Regular grammars, finite automata
- **Context-Free** — CFGs, pushdown automata
- **Context-Sensitive** — CSGs, linear-bounded automata
- **Recursively Enumerable** — Unrestricted grammars, Turing machines

[2] E.g., "kamusta" $\in \mathcal{F}$ but "asdfgh" $\notin \mathcal{F}$; thus $\mathcal{F} \subseteq \Sigma^*$

Some rules are regular

Object-Focused Future Tense in Bikol

Object-Focused Future Tense (OFFT) verbs in Bikol add suffixes to base verbs.

Rule: Future tense verb with suffix determined by base verb ending:

- Ends with **katinig** (consonant) ⇒ suffix "-on"
- Ends with **patinig** (vowel) ⇒ suffix "-hon"

Examples:

- Gigibo**hon** (to give) — *gibo* ends in vowel
- Apod**on** (to clean) — *apod* ends in consonant

Naive Regular Grammar

Let $\mathbb{VERB}$ = set of all valid Bikol verbs.

Attempt 1: Simple regular grammar

$$
\begin{aligned}
S &\to \mathbb{VERB}\ \text{hon} \mid \mathbb{VERB}\ \text{on} \\
\mathbb{VERB} &\to \text{gibo} \mid \text{apod} \mid \ldots
\end{aligned}
$$

Problem: This doesn't account for verb endings!

It would incorrectly accept "giboon" and "apodhon"

Feature Tagging to the Rescue

Solution: Partition verbs by their ending

- $\mathbb{VERB}_p$ — verbs ending in **p**atinig (vowel)
- $\mathbb{VERB}_k$ — verbs ending in **k**atinig (consonant)

Properties:

- $\mathbb{VERB}_p \cap \mathbb{VERB}_k = \emptyset$ (disjoint)
- $\mathbb{VERB}_p \cup \mathbb{VERB}_k = \mathbb{VERB}$ (complete)

This technique is called **feature tagging**!

Corrected Regular Grammar

With feature tagging:

$$S \to \mathbb{VERB}_p \text{ hon} \mid \mathbb{VERB}_k \text{ on}$$
$$\mathbb{VERB}_p \to \text{gibo} \mid \ldots$$
$$\mathbb{VERB}_k \to \text{apod} \mid \ldots$$

Result:

- Produces *only* valid OFFT verbs
- Verification in $\mathcal{O}(n)$ time [3]

[3] See [Tho68] for $\mathcal{O}(n)$ regex matching; also RE2 by Russ Cox et al. at Google: `https://github.com/google/re2`

When Regularity Works

Key insight: We can *coerce* regularity through:

- Feature tagging
- Inflating the symbol space

This works here because:

- Only one non-terminal needs splitting
- The split is binary (vowel vs. consonant)

But what happens when things get more complex?

Enclitic Particles in Tagalog

Enclitic Particles (EP) add meaning to preceding words.

Examples:

- "rin" / "din" — inclusion ("too", "also")
- "raw" / "daw" — reported speech ("supposedly")

Morphophonemic Rule:

- Use **r-form** if preceding word ends in vowel (patinig)
- Use **d-form** if preceding word ends in consonant (katinig)

Examples: Reported Speech

Correct usage:

1. Marami **raw** — ends in vowel *i*
2. Humingi **raw** — ends in vowel *i*
3. Sinapak **daw** — ends in consonant *k*
4. Natulog **daw** — ends in consonant *g*

Can We Use Feature Tagging?

Reported speech EPs can follow multiple parts of speech:

- Nouns ($\mathbb{NOUN}$)
- Verbs ($\mathbb{VERB}$)
- Adjectives ($\mathbb{ADJ}$)
- Adverbs ($\mathbb{ADV}$)

Yes! Similar to section 1, we can construct a regular grammar with feature tagging.

Regular Grammar with Feature Tagging

$$
\begin{aligned}
S &\to \mathbb{NOUN}_p \text{ raw } \mid \mathbb{NOUN}_k \text{ daw} \\
&\mid \mathbb{VERB}_p \text{ raw } \mid \mathbb{VERB}_k \text{ daw} \\
&\mid \mathbb{ADJ}_p \text{ raw } \mid \mathbb{ADJ}_k \text{ daw} \\
&\mid \mathbb{ADV}_p \text{ raw } \mid \mathbb{ADV}_k \text{ daw} \\
\mathbb{NOUN}_p &\to \text{Lola} \mid \ldots \\
\mathbb{NOUN}_k &\to \text{Juan} \mid \text{Bert} \mid \ldots \\
\mathbb{VERB}_p &\to \text{Humingi} \mid \text{Umiiyak} \mid \ldots \\
\mathbb{VERB}_k &\to \text{Kumain} \mid \text{Sumigaw} \mid \ldots \\
\mathbb{ADJ}_p &\to \text{Masaya} \mid \text{Maganda} \mid \ldots \\
\mathbb{ADJ}_k &\to \text{Matulin} \mid \text{Mabilis} \mid \ldots \\
\mathbb{ADV}_p &\to \text{Ngayon} \mid \text{Dati} \mid \ldots \\
\mathbb{ADV}_k &\to \text{Noon} \mid \text{Hapon} \mid \ldots
\end{aligned}
$$

The Problem: Combinatorial Explosion

Issue: Feature tagging creates production explosion!

For n POS tags, each needing vowel/consonant split:

$$\text{Total productions} = 2n + 1$$

Consequence:

- Grammar becomes unwieldy
- Difficult to maintain
- Defeats the purpose of "easy to comprehend"

So…Is It Regular?

Theoretically: Yes! Morphophonemic alternation *is* regular.

Pragmatically: Not really!

The rule-based philosophy:

- Leverage human comprehensibility
- Keep grammars maintainable
- Avoid spatial/practical inefficiency

A spatially inefficient rule-based approach is a (essentially) **useless**.

Context Sensitivity

The Wild West of Grammars

Context-Sensitive Grammars (CSGs) are powerful:

General CSG production form:

$$xSy \to x\Phi y, \quad \Phi \neq \epsilon$$

Non-terminal S rewrites to Φ in context $x \cdots y$

Our focus: Attribute Grammars (a specific type of CSG)

Attribute Grammars

Attribute Grammar (AG): CSG where each symbol has associated attributes

Intuition:

- Start with a CFG
- Add rules for computing attribute values
- Values depend on context

Common uses:

- Semantic analysis in compilers
- Type checking
- Code generation

Morphophonemic Alternation is regular but AGs are a better represent it

Recall morphophonemic alternation from section 2.

Instead of feature tagging (which explodes),
use an attribute grammar to handle context sensitivity:

$$
\begin{aligned}
S \to \Phi \text{ raw} \quad & : \Phi \in \mathbb{NOUN} \cup \mathbb{VERB} \cup \mathbb{ADJ} \cup \mathbb{ADV},\ \text{last}(\Phi) \in \{\text{vowels}\} \\
\mid \Phi \text{ daw} \quad & : \Phi \in \mathbb{NOUN} \cup \mathbb{VERB} \cup \mathbb{ADJ} \cup \mathbb{ADV},\ \text{last}(\Phi) \notin \{\text{vowels}\}
\end{aligned}
$$

where $\text{last}(\Phi) = \Phi_{|\Phi|-1}$.

Key: The condition checks context (last character) without exploding the grammar.

Benefits of Attribute Grammars:

- **Compactness:** Avoids combinatorial explosion of productions
- **Maintainability:** Easier to update rules without rewriting large parts
- **Clarity:** Clearly separates structure from context-dependent rules

Trade-off: AGs in **PSPACE-complete**!

Habitual Action in Tagalog is *Midly* Context-Sensitive

Habitual Action denotes that a verb is done regularly.

Rule: enclose the word “nang” with a reduplication of the verb.

Examples:

1. Kain **nang** kain
2. Tulog **nang** tulog

Habitual Action is at least Regular

All finite (a union of singletons) languages L are regular (think of this as brute forcing all singletons as a production from the start symbol).

So, since the set of verbs $\mathbb{VERB} = \{v_0, v_1, \ldots, v_n\}$ is finite, the habitual action language is also regular.

Habitual Action is Not Practical as Regular

But again, this is not a practical grammar! It requires enumerating all verbs in the language, which is finite but very massive!

Would implementing a CFG be any better?

Habitual Action is *Still* Not Practical as Context-Free

Since all finite languages are regular, they are also context free (**REG** $\subseteq$ **CFG**).

But this still requires enumerating all verbs in the language, which is finite but very massive!

$$S \to v_0 \text{ nang } v_0 \mid v_1 \text{ nang } v_1 \mid \ldots \mid v_n \text{ nang } v_n \mid$$

Habitual Action = Copy Language

If we consider the $\mathbb{VERB}$ set to be infinite, the language **HA** $= \{v \text{ nang } v \mid v \in \mathbb{VERB}\}$ can be seen as analogous to the **Copy-Language** $= \{ww \mid w \in \Sigma^*\}$. From this, it follows that the habitual action construction is context-sensitive.

Habitual Action is Midly Context-Sensitive

Mildly Context-Sensitive Languages (**MCSLs**) are a subclass of context-sensitive languages, they occupy the boundary between **CFL** and **CSL**s,[4].

$$
\begin{aligned}
\mathbf{S} &\to X \text{ “nang” } Y && \{\, X.\texttt{word} = Y.\texttt{word} \,\} \\
X &\to \text{“kain”} \mid \text{“tulog”} \mid \ldots \\
Y &\to \text{“kain”} \mid \text{“tulog”} \mid \ldots
\end{aligned}
$$

Remark. You could remove the Y non-terminal and just have X generate the verb again, but this is to illustrate the copying constraint more clearly.

[4]They can be thought of as the simplest or least complex members of the set of **CSL**—complex enough to exceed the expressive power of **CFLs**, yet not as *hard* as the more general **CSLs**.

Advanced Concepts and Future Directions

Languages and Decision Problems

Decision Problem (DP): A yes/no question

Every formal language $\mathcal{L}$ defines a decision problem:

$$\text{Given } s \in \Sigma^* \text{ and } \mathcal{L} \subseteq \Sigma^*, \quad \text{is } s \in \mathcal{L}?$$

This is the **membership query**.

Complexity Classes

Complexity Classes categorize problems by computational resources needed.

Classes of interest:

- **P** (Polynomial Time) — Solvable in polynomial time
- **PSPACE** (Polynomial Space) — Solvable with polynomial memory

Language Complexity

Time complexity by language class:

- **Regular** — $\mathcal{O}(n)$ (subset of **P**)
- **Context-Free** — $\mathcal{O}(n^m)$ (subset of **P**)
- **Context-Sensitive** — **PSPACE**
- **Recursively Enumerable** — Unbounded

Key Takeaway: Higher in the hierarchy = more expressive, more expensive to process

Intractability

This work shows that Tagalog is *at least* context-sensitive due to the existence of the HA rule.

Proving wether Tagalog (or any language in general) is (midly) context-sensitive is an open problem.

Hence, parsing Tagalog is an intractable problem leaving room for approximation techniques [5].

[5] See [SA24, Bra18]

There are a lot of rules in Tagalog grammar that are not yet formalized. This work only covers a subset of the grammar. We may formalize it manually, but this is

time-consuming and labor-intensive. **Grammar Induction** is the task of learning

grammar rules from a corpora (you can use the Bible corpus you got for the first project!) [6].

[6] See [Ang87, ZFW+25, WGY22].

General reference: [GS21, AsWF14, sWF14, Mal09, Min19].

Dana Angluin.
Learning regular sets from queries and counterexamples.
Information and Computation, 75(2):87–106, 1987.

Virgilio Almario and Komisyon sa Wikang Filipino.
KWF Manwal sa Masinop na Pagsulat.
Komisyon sa Wikang Filipino, 2014.

António Branco.
Computational complexity of natural languages: A reasoned overview.
In Leonor Becerra-Bonache, M. Dolores Jiménez-López, Carlos Martín-Vide, and Adrià Torrens-Urrutia, editors, *Proceedings of the Workshop on Linguistic Complexity and Natural Language Processing*, pages 10–19, Santa Fe, New-Mexico, August 2018. Association for Computational Linguistics.

Kyle Gorman and Richard Sproat.

***Finite-State Text Processing*.**
Springer International Publishing, 2021.

Valeria Malabonga.
Heritage voices: Language - tagalog.
Heritage Voices Collection, January 2009.

Malcolm W Mintz.
***Bikol Dictionary*.**
University of Hawaii Press, March 2019.

Hajime Senuma and Akiko Aizawa.
Computational complexity of natural morphology revisited.
Transactions of the Association for Computational Linguistics, 12:649–663, 2024.

Komisyon sa Wikang Filipino.
***Ortograpiyang Pambansa*.**

Komisyon sa Wikang Filipino, 2014.

Ken Thompson.
Programming techniques: Regular expression search algorithm.
Commun. ACM, 11(6):419–422, June 1968.

Gail Weiss, Yoav Goldberg, and Eran Yahav.
Extracting automata from recurrent neural networks using queries and counterexamples (extended version).
Machine Learning, 113(5):2877–2919, 2022.

Yu Zhao, Hao Fei, Shengqiong Wu, Meishan Zhang, Min Zhang, and Tat seng Chua.
Grammar induction from visual, speech and text.
Artificial Intelligence, 341:104306, 2025.