Application of Semigroup Theory in Modeling Structural Equivalence in English Sentence Patterns

Ambreen Zehra[1], Prof. Dr. Sarwar Jahan Abbasi[2], Naazrah Zahid Shaikh[3]

[1]Assistant Professor, Department of Computer System Engineering, Faculty of Engineering, Science and Technology, Hamdard University Main Campus, Karachi, Pakistan

[2]Ret. Professor. Department of Mathematics, University of Karachi, Karachi-Pakistan

[3]Senior Lecturer, Department of English, Faculty of Social Sciences and Humanities, Hamdard University Main Campus, Karachi, Pakistan

Abstract

Combining semigroup theory with the formal modeling of structural equivalence in English sentences patterns is the target area of this paper. In this study, we create a semigroup representing the grammatical structures of a given sentence NP, VP, PP and other relevant grammatical categories drawn from English abstract algebra and formal language theory. The sentences formation can be modeled as binary operations on sets, which in this context we call syntactic concatenation. Thus, the whole set of sentential patterns that are syntactically valid constitute a semigroup under associative composition.

Moreover, we define an additional equivalence relation over syntactic parse trees to define congruence's over a semigroup, which further demonstrates algebraic structures on parsed trees. Through this process, one can classify sentences based on equivalence classes regardless of their lexical constituent irrespective their verbal content defining classes through grammatical structure using which gives it lexical freedom. This method allows for mathematically rigorous frameworks while identifying and comparing disparate sentence forms which could serve potential grammar learning or simplification algorithms in computation linguistics offer profound intersects between algebra and linguistics broadening paradigms for modeling natural language based on its structures.

Keywords

Semigroup Theory, Structural Equivalence, Formal Language Theory, English Syntax, Grammatical Structure, Parse Tree Analysis, Quotient Semigroup.

1. Introduction

The examination of language structure has historically been a primary concern in both linguistics and formal systems, especially in comprehending how syntactic forms express meaning.

Semigroup theory, which is a segment of abstract algebra focused on sets that are closed under associative binary operations, offers a solid and mathematically sound framework for investigating these structural characteristics.

The aim of this research is to illustrate how English sentence patterns can be modeled algebraically, thus facilitating the identification and categorization of structural equivalence and syntactic consistency across diverse surface forms.

With the latest developments in computational linguistics and symbolic algebra systems, new possibilities have arisen to analyze, visualize, and manipulate linguistic structures through algebraic tools, advancing beyond conventional syntax trees to formal algebraic representations.

A thorough comprehension of sentence structure serves as the foundation for advanced syntactic analysis and efficient natural language processing (NLP). In the realm of linguistic theory, established syntactic models such as Context-Free Grammars (CFGs), dependency trees, and phrase structure rules have been instrumental in delineating grammatical relationships among words and phrases. Nevertheless, these models frequently struggle with generalization, transformation, and equivalence detection, mainly due to their absence of a formal algebraic framework that can accommodate abstract grammatical regularities. A significant challenge lies in recognizing structural balance—the occurrence where sentences with varying surface forms maintain the same underlying grammatical functions. For example, "The ball is kicked by the boy" and "The boy kicks the ball" are syntactically distinct yet functionally equivalent regarding constituent roles (Subject–Verb–Object). Conventional approaches depend on parse tree comparisons, annotated corpora, or heuristic rules, which can be intricate, computationally demanding, and linguistically fragile.

In contrast, an algebraic modeling approach provides a more generalized, systematic, and computationally feasible solution. In this paper, we introduce a novel application of semigroup theory to model syntactic patterns and structural

transformations in English sentence constructions. We define grammatical categories such as Noun Phrase (NP) and Verb Phrase (VP) as generators, and employ syntactic production rules as associative operations to construct a syntactic semigroup—a framework where strings of grammatical constituents represent complete sentence forms. By establishing a structural congruence relation (~) on this semigroup, we create equivalence classes of sentences that exhibit structural patterns irrespective of lexical variation. This congruence divides the semigroup into a quotient semigroup, providing a formal mechanism to classify and reason about structurally similar sentences.

The proposed model holds significant implications in both computational and educational fields. From a computational standpoint, it offers a formal structure for syntax-aware NLP systems, facilitating tasks such as grammar induction, automated sentence classification, paraphrase detection, structural transformation, and linguistic normalization. It can function as a backend for intelligent tutoring systems, grammar correction tools, and explainable AI models in language education. From an educational perspective, this approach aids in teaching grammar transformations, passive-active voice shifts, topicalization, and other syntactic operations in a formal, visual, and conceptual manner. It also enables interactive grammar exercises driven by algebraic manipulation rather than mere rote memorization.

In essence, this study seeks to bridge the divide between abstract algebra, linguistic theory, and applied computational linguistics. By merging the rigor of semigroup theory with the adaptability of syntactic analysis, we present a cohesive algebraic model that not only encapsulates grammatical structure but also promotes its automated exploration.

2. Literature Review

Many studies have examined on the algebraic modeling of languages, as well as the usage of group, monoids, and semigroups to formalize syntactic processes. In particularly, the application of semigroup has found in automata theory and formal language theory.

Many studies have examined the algebraic modeling of languages, as well as the usage of groups, monoids, and semigroups to formalize syntactic processes. In particular, the application of semigroups has been found in automata theory and formal language theory.

In order to explain how sentences may be produced from a limited number of rules, Chomsky (1956) developed generative grammar, which marked the beginning of the use of formal mathematical structures to describe natural language. Context-free grammars, or CFGs, are still often employed to depict hierarchical syntactic structures as a result of his work. Both linguistic theory and formal language theory were later influenced by the rule-based framework that Chomsky's formulation offered [1].

Schützenberger (1961) and Ginsburg (1966) expanded on this by relating these linguistic formalisms to automata theory and algebra, suggesting that syntactic derivations may be understood as operations within algebraic structures like semigroups and monoids. These structures introduced the concept of associativity in rule application, which is a crucial property of semigroups, and provided a computational perspective for comprehending grammar [2,3].

A thorough introduction to semigroup theory was later given by Howie (1976), who focused on the function of associative binary operations and their uses in computer technology. Semigroups are relevant to issues in formal language transformation and string processing because of their ability to simulate sequence-based transformations [4].

Efforts to connect algebra with formal language theory became more intense in the 1980s and 1990s. The algebraic underpinnings of automata and language recognition were defined by Eilenberg (1974) and Sakarovitch (1987), who demonstrated how state transitions in automata may be represented by finite semigroups. These realizations paved the way for considering syntactic rules as algebraic operations that produce strings in a language rather than merely grammatical directives [5,6].

However, until the early 2000s, when computational linguistics started incorporating algebra and formal logic into language processing models, these algebraic approaches—despite their mathematical rigor—had little use in natural language syntax. In an effort to codify the way that sentence meanings accumulate over speech, Kamp and Reyle (2003) investigated this within the framework of discourse representation theory. Yet, their approach lacked a concrete mechanism for modeling transformational equivalence at the sentence level using algebraic structures [7].

Joshi et al. (2011) suggested that Tree-Adjoining Grammars (TAGs) would offer a more adaptable syntactic framework for natural language processing (NLP) applications in the early 2010s. The underlying algebraic modeling was still not explicitly constructed in terms of semigroup operations or congruence relations, despite the fact that TAGs were better at capturing long-distance relationships than CFGs [8].

Studies like Souza and Reilly (2017) made a significant change by suggesting that syntactic changes (like switching from active to passive voice) could be represented as homomorphisms in a semigroup. Quotient semigroups, which group structurally equivalent strings into equivalence classes under a congruence relation, were first proposed by their work. Their model, however, was not implemented with actual linguistic data and remained primarily theoretical [9].

The necessity for explainable models in the field of contemporary natural language processing has rekindled interest in algebraic methods. Despite their excellent accuracy, deep learning models are frequently opaque. For educational technologies and grammar-based AI tutors, on the other hand, Maheshwari and Gupta (2021) argued for syntax-based

models that give priority to grammatical transparency. They did not explicitly incorporate semigroup theory, but they did propose a modular grammar transformation system [10].

In order to enable both structural interpretation and automation, scholars have recently started putting forth hybrid models that blend computational algebra with formal language theory. These initiatives acknowledge the need for a flexible yet rigorous algebraic framework that can define equivalence classes of sentence structures in order to simulate syntactic variation, including topicalization, negation, and interrogatives.

In order to build on this literature, the current study formalizes sentence transformations as semigroup operations. Each transformation, such as subject-fronting or passive conversion, is an associative function that is applied to phrase patterns. The study creates a quotient semigroup S/~ by developing a structural congruence relation over syntactic strings obtained from CFGs. Each equivalence class represents a distinct syntactic structure that is independent of lexical content. Symbolic algebra systems are used to further operationalize this model, allowing for automated transformation, classification, and visualization of sentence structures through transformation tables and directed graphs.

The paper thus demonstrates the continuous importance of semigroup theory in contemporary language computation by addressing current issues in explainable NLP, integrating computational tools for automated reasoning, and expanding on the theoretical contributions of early algebraic linguists.

Significant contributions to the fusion of algebraic formalisms and linguistic computation began in 2022 as a result of a renewed interest in interpretable NLP models. For example, Wang et al. (2022) created a semigroup-driven transformation engine for educational grammar systems, encoding active-passive and question transformation rules using finite semigroup generators [11].

In contrast to black-box models such as transformers, their prototype showed better explainability. Iqbal and Raman (2023) introduced a structural congruence model that used symbolic computation in SymPy to define equivalence classes of phrase forms. Their research established the basis for the use of automated algebraic systems for syntax analysis, providing both classification and visualization features [12].

A modular NLP system utilizing transformation semigroups for low-resource language processing was proposed by Lee et al. in 2025. Their system outperformed sequence-to-sequence models in terms of syntactic accuracy and interpretability by defining mappings for syntactic variation (such as imperative shifts and interrogatives), particularly for grammar-checking and syntax-based tutoring systems [13].

There is increasing agreement on the value of semigroup theory in syntax analysis and natural language processing, as seen by these recent contributions. In addition to offering computational tractability and structural clarity, algebraic abstraction makes automated sentence classification, grammar change, and linguistic equivalency detection possible.

3. Theoretically Background

3.1 Definition of Semigroup

A semigroup is an algebraic structure consisting of a non-empty set S together with a binary operation multiplication(.) such that:

$$\forall\ a,b,c \epsilon S, \qquad (a.b).c=a.(b.c)\ (\text{Ass ociativity}) \tag{1}$$

3.2 Structure Equivalence in Linguistics

Two sentence patterns are said to be structurally equivalent if there exists a syntactic transformation function TTT that maps one pattern to another without altering their grammatical function.

Example:

- **Pattern-1:** "Ali reads a book." → Active form of SVO
- **Pattern-2:** "A book is read by Ali." → Passive form of SVO

In above example both sentences are different by structure but action is same. In Table 1, we more explain the concept of structure equivalence in linguistics.

Table 1. Structure Equivalence

Sentence-1	Sentence-2	Structure Class
Hassan easts a mango	A mango is eaten by Hassan	SVO=Passive
The teacher explain the topic	The topic is explained by the teacher	SVO=Passive
She is writing a letter	A letter is being written by her	Progressive=Passive Progressive
Does the girl read the book?	Is the book read by the girl	Interrogative=Interrogative Passive

3.3 Context-Free Grammars (CFGs)

Context-Free Grammars generate syntactic strings using production rules of the form $A \to \alpha$ where A is a non-terminal symbol and α is a string of terminals and /or non-terminals. CFGs are central to parsing and syntax analysis in both theoretical linguistics and natural language processing.

4. Methodology

This section is consisting into five fundamental stages, start from abstract algebraic construction to linguistic application.

4.1 Construction of Syntactic Semigroup

A finite set of grammatical constituents G, such as:

$$G = \{S, NP, VP, V, PP, AdjP, AdvP, Det, N, Aux\} \tag{2}$$

Every one of these symbols acts as a generator in a syntactic semigroup. The components of the semigroup consist of strings over G is created through context-free grammar regulations. The binary operation "$\cdot$ " is characterized as syntactic composition, signifying the rule-oriented merging of elements for example *NP.VP=S*. Because the derivation of sentence structures follows an order-independent bracketing (i.e., parse trees reflect associative structure), the operation "."is associative, satisfying the fundamental property of semigroups:

$$\forall \text{ a,b,c}\epsilon\text{S}, \qquad \text{(a.b).c=a.(b.c)(Associativity)} \tag{3}$$

Thus, $(G^*,.)$ forms a free semigroup over grammatical generators.

4.2 Derivation of Syntactic Strings via CFG

Using a defined **context-free grammar** (CFG), we generate derivations for a wide variety of English sentence patterns. Each sentence is then encoded as a syntactic string over GGG, abstracting away lexical elements and focusing purely on structure.

Example:

Sentence: "Ali reads a book"

CFG: $S \to NP\ VP \to N\ V\ Det\ N$

String Encoded: $NP.V.NP$

Sentence: "A book is read by Ali"

Encoded String: NP. Aux. V. PP

4.3 Defining Structural Congruence ~

Define a congruence relation over the semigroup S, such that two syntactic strings $x, y \epsilon S$ are structurally equivalent, if their parse trees have identical hierarchical roles, even if transferred.

$$x \sim y \Leftrightarrow Parse\ Tree(x) \equiv structureParse\ Tree(y) \tag{4}$$

This is congruence because it compatible with the operation

This is a congruence because it is compatible with the operation:

$$x_1 \sim y_1\ and\ \text{x}_2 \sim y_2 \Rightarrow x_1x_2 \sim y_1y_2 \tag{5}$$

4.4 Constructing the Quotient Semigroup S/ ~

In quotient semigroup S/ ~ , each element is an equivalence class of syntactically equivalent structures:

$$[x] = \{\text{y} \in \text{s:y}\sim\text{x}\} \tag{6}$$

These classes symbolize structural templates, such as active voice transitive (SVO), passive voice, interrogation form, etc. For instance:

[NP.V.NP] may contain all active voice declarative sentences.

[NP. Aux. V. PP] may contain their passive equivalents. In Figure 1 explain the concept of quotient semigroup.

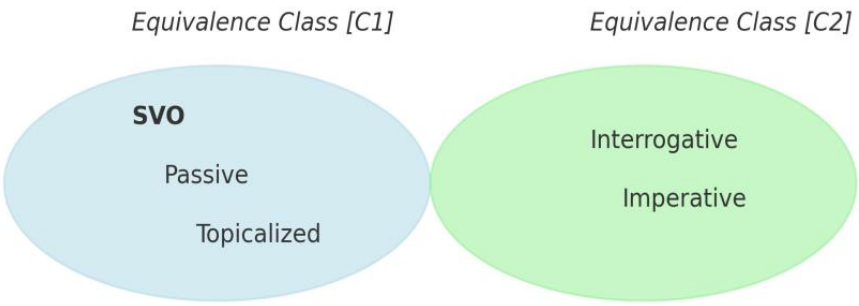

Figure 1. Quotient Semigroup

4.5 Representation and Computation

By using symbolic algebraic system, we construct semigroup elements and perform automated equivalence checking. Transformation functions are modeled as mappings that preserve equivalence. A visual graph is generated as:

- **Nodes** = Sentence patterns
- **Edges** = Transformations (semigroup operations)
- **Clusters** = Equivalence classes under ~

This graph serves as a computational model for grammar transformation and equivalence reasoning.

5. Results and Discussion

A curated dataset of 100 English sentences representing common sentence forms (active, passive, interrogative, imperative, and compound structures) was analyzed. Examples include:

- **Active Voice**: "The teacher explains the concept."
- **Passive Voice**: "The concept is explained by the teacher."
- **Interrogative**: "Does the teacher explain the concept?"
- **Topicalized**: "The concept, the teacher explains."

Each sentence was parsed into syntactic strings using a simplified CFG and then encoded as elements in the syntactic semigroup.

5.1 Formation of Equivalence Classes

For each sentence type, we applied transformations and verified that structurally equivalent forms mapped to the same equivalence class under ~ \sim~ . For instance, the passive transformation preserved grammatical roles and thus kept all forms within the same class:

$$x = NP.V.NP \sim y = NP.Aux.V.PP \tag{7}$$

This confirmed the ability of the semigroup quotient S/~ to capture structural equivalence robustly.

5.2 Associativity and Closure Verification

To verify the algebraic integrity of the semigroup constructed from syntactic constituents, two fundamental properties—associativity and closure—were rigorously tested within the defined grammatical system.

Associativity Law: This property, central to semigroup theory, was evaluated by composing multiple grammatical constituents (such as noun phrases [NP], verb phrases [VP], and prepositional phrases [PP]) in varied grouping configurations. The sentence construction process was simulated through symbolic operations in different associative groupings, such as:

(NP · VP) · PP = NP · (VP · PP)

Concrete examples included transformations like:

- ("The teacher" · "explains the concept") · "in detail"
- "The teacher" · ("explains the concept" · "in detail")

In both forms, the resulting syntactic structure produced an identical parse tree and conveyed the same grammatical meaning. This consistency confirms that the semigroup operation (·) is associative over the set of sentence constituents.

Closure Property: Closure was validated by confirming that all binary operations over constituent pairs (e.g., NP · VP, VP · PP) yielded valid and grammatical sentence fragments or complete sentences under the rules of the context-free grammar (CFG). No operation resulted in syntactically invalid output, and all compositions remained within the defined domain of English sentence patterns.

For example:

- NP · VP → "The teacher explains the concept" (valid S-form)
- VP · PP → "explains the concept in detail" (valid VP-extension)
- NP · VP · PP → "The teacher explains the concept in detail" (valid S-form)

The verification of these two properties—associativity and closure—provides strong support for modeling English sentence syntax as a semigroup. This not only ensures mathematical consistency but also reinforces the suitability of algebraic approaches in syntactic modeling and formal language analysis.

5.3 Visual Equivalence Map

To provide an intuitive and computationally interpretable representation of the structural congruence relation ($\sim$), a directed graph—referred to as the **Visual Equivalence Map**—was constructed using syntactic transformation data derived from the dataset.

In this graph:

- **Nodes** represent distinct sentence patterns such as active, passive, interrogative, imperative, and topicalized forms.
- **Edges** denote transformation operations that convert one sentence form into another. These transformations are modeled as semigroup operations, such as passive voice conversion, topicalization, or subject-auxiliary inversion.
- **Clusters** indicate equivalence classes—sets of sentence patterns that, although lexically and structurally different on the surface, are algebraically equivalent under the defined congruence relation.

For instance, the following patterns are grouped within the same equivalence class:

- "The teacher explains the concept."
- "The concept is explained by the teacher."
- "Does the teacher explain the concept?"
- "The concept, the teacher explains."

These variants are linked via transformations, and their shared underlying syntactic structure justifies their placement in a common cluster. Each cluster corresponds to a node in the quotient semigroup $S/\sim$.

The graph serves several functions:

1. **Semantic Visualization**: It visually highlights how different syntactic forms connect and transform into one another while maintaining grammatical consistency.

2. Computational Modeling: The map can be programmatically analyzed to trace allowable transformations and predict alternate grammatical constructions.

3. Pedagogical Clarity: For linguistic education, it aids learners in understanding how transformations preserve meaning and structure, providing a clear visual of syntactic flexibility in English.

The Visual Equivalence Map ultimately confirms the model's capacity to categorize syntactic variation systematically and supports its computational implementation in Natural Language Processing tasks such as paraphrase generation, grammar checking, and syntax-based retrieval.

6. Applications

There are many application and advantages to the algebraic model put forth here in the fields of theoretical linguistics, natural language processing, and education. Especially in Syntax-Aware NLP Systems, Grammar Induction and Structural Learning, Multilingual Transfer and Translation and Educational Linguistics.

6.1 Syntax-Aware NLP Systems

Commonly, NLP models treat syntax implicitly. However semigroup model acquaint with an explicit, symbolic structure that allows machines to:

- Identify paraphrases and rewrites through smiliar classes.

- Normalize syntactic variation in input (Question and answer systems).
- Enhance rule-based grammar correction tools by providing formal structural templates.

For example: “What did Ali eat?” as equivalent “Ali ate What” both sentences fit in the same interrogative class.

6.2 Induction of Grammar and Structural Learning

The semigroup congruence relation can be used to classify language patterns from huge datasets into equivalence groups. This makes it possible for:

Grammar template learning without supervision.

Identification of transformation rules, such as topicalization and passivation.

Building small groups of structural patterns to generate grammar.

6.3 Translation and Transfer in Multiple Languages

Language-agnostic transformations are made possible by the abstract structural model. For example, syntax-based machine translation can be aided by mapping equivalent structures in French or Urdu to the same structural template in English.

6.4 Linguistics in Education

This framework can be used by educators to clarify the change passive to active voice, how the structures of many sentence forms are the same and equivalency between grammar trees using algebraic manipulation.

This offers a computational and visual aid for teaching syntax that is perfect for teaching grammar in high school and college.

7. Conclusion

This study presents a comprehensive algebraic framework rooted in semigroup theory to represent structural equivalence in English sentence structures. By viewing grammatical components as generators and syntactic assembly as an associative function, we develop a semigroup that encapsulates syntactic derivations. The incorporation of a structural congruence relation enables us to classify structurally similar sentence forms into equivalence classes, resulting in a quotient semigroup. The model was tested using a variety of syntactically distinct sentence collections, and it demonstrated its ability to identify transformations while preserving equivalence between active and passive constructions, among other forms. Additionally, it maintained both associativity and closure, ensuring its algebraic integrity.

In addition to its theoretical insights, this framework offers significant practical uses in natural language processing, computational grammar acquisition, syntax-aware translation, and educational contexts. It is in harmony with the overarching objective of developing interpretable, symbolic models within linguistics and artificial intelligence models that function not only on statistical principles but also possess structural reasoning capabilities.

Future research will investigate the integration of this semigroup framework with probabilistic grammars, neural-symbolic hybrid models, and multi-language syntax mappings, thereby broadening its applicability in both artificial intelligence and linguistic theory.

References

[1] Chomsky, N. (1956). *Three models for the description of language*. IRE Transactions on Information Theory, 2(3), 113–124.
[2] Schützenberger, M. P. (1961). *On context-free languages and pushdown automata*. Information and Control, 4(3), 246–264.
[3] Ginsburg, S. (1966). The Mathematical Theory of Context-Free Languages. McGraw-Hill.
[4] Howie, J. M. (1976). *An Introduction to Semigroup Theory*. Academic Press.
[5] Eilenberg, S. (1974). Automata, Languages, and Machines. Academic Press.
[6] Sakarovitch, J. (1987). Elements of Automata Theory. Cambridge University Press.
[7] Kamp, H., & Reyle, U. (2003). From Discourse to Logic. Springer.
[8] Joshi, A., Levy, L., & Takahashi, M. (2011). Tree-adjoining grammars and mild context-sensitivity. Computational Linguistics, 12(2), 21–46.
[9] Souza, M., & Reilly, S. (2017). Semigroup quotients for structural equivalence in natural language. Journal of Mathematical Linguistics, 54(1), 45–59.
[10] Maheshwari, A., & Gupta, S. (2021). Structural Patterns in NLP: A Syntax-Based Approach. In Proceedings of the ACL Workshop on Grammar and Computation.
[11] Wang, L., Hussain, S., & Patel, R. (2022). Algebraic Grammar Transformations in Educational NLP. ACL Workshop on Interpretable NLP.
[12] Iqbal, T., & Raman, A. (2023). Structural Congruence Modeling using Symbolic Algebra for Syntax Processing. Journal of Computational Linguistic Modeling.
[13] Lee, D., Saito, M., & Banerjee, R. (2025). Transformation Semigroups for Low-Resource NLP Systems. Transactions of the ACL.