A Pregroup Representation and Analysis of Hindi Syntax

Thesis submitted in partial fulfillment
of the requirements for the degree of

*(Master of Science in **Computational Linguistics** by Research)*

by

Alok Debnath
20161122
alok.debnath@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2021

Copyright © Alok Debnath, 2021
All Rights Reserved

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "A Pregroup Representation and Analysis of Hindi Syntax" by Alok Debnath, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Dr. Manish Shrivastava

To l’arbitraire du signe

Acknowledgments

This masters thesis has been in the works since the summer of 2017. A paper published in 2010, "The Mathematical Foundations of Compositional Distributional Models of Meaning" coauthored by Bob Coecke, caught my attention for its unique theoretical take on compositional semantics and combining it with distributionality. Over the years that passed, I wanted to make a representation of that form for Hindi as well, where a uniform syntactic representation could be ascertained from that notion.

Nearly four years and two publications after, I present to the world a small formalization of Hindi syntax for the purpose of exploring compositional semantics for languages with unique syntactic constraints. During the process, through the ups and downs, pitfalls and frustrations of formalizing Hindi syntax, a group of people silently cheered me on, and I could not have done this without them. I am indebted to my supervisor, Dr. Manish Shrivastava, for allowing me to pursue this blatantly theoretical path and helping me realize a fraction of my potential. His faith in me kept me going in times my faith in myself waned and withered. I'd also like to thank my parents, Dr. Somnath Debnath and Jyoti Debnath, for always being by my side, through the pains and pleasures of research and publications. Thank you so much.

A number of researchers played a huge role in keeping me motivated. Professor Bob Coecke, whose brains I picked to realize that it was actually possible to work on this topic for Hindi syntax, Professors Claudia Casadio and Mehrnoosh Sadrzadeh for their valuable insights at a time when I did not understand the relevance or significance of their words or their work, and Professor Marie-Catherine de Marneffe, for mentoring me through the process of paper writing for my first first-authored publication at NAACL's Student Research Workshop in 2019. Your help provided the asphalt for me to pave the way to write this thesis, and much beyond.

Closer to home, numerous phone calls and messages of frustration exchanged between friends and close ones without whom a large part of this would not have been possible. Thank you Nitin, Siddharth, Projit, and Anshita for being enthusiastic moral support in my success, and a helping hand in my failures. Much of this thesis would have remained unwritten if not for your presence. In a similar vein, Saujus and Mukund, even into the final days of writing, your helped clarify and crystallize concepts that would have remained blurry under the stress of compiling all this work into something worth writing. There have been so many other people whose words and work contributed to me being able to work on this topic and research in general, I apologize for not naming all of you. Your contributions are not forgotten,

and I am deeply grateful to each and every person I encountered in my research path, who helped me, motivated me, and guided me to be able to do research on something I liked and enjoyed.

Thank you.

Abstract

In 1957, Noam Chomsky's revolutionary treatise, *Syntactic Structures* [45], established the foundations for a formal study of natural language syntax. Such a study requires the development of a mathematical paradigm where the grammatical structure of the language can be analyzed sans the semantic conditions responsible for the "possibility" or truth-value of the sentence, hence *colourless green ideas sleep furiously*. The work of Chomskyian linguistics in differentiating shallow and deep structures perpetuated the ideas of a rigid motif of structural integrity to language, which was often challenged by semanticists. However, it established the path of the study of natural language by the medium of syntax, therefore impressing upon the need to understand the sentence structure as a method of analyzing the language in a typologically uniform and relevant manner.

In this thesis, we study the computational and representational offshoot of syntactic structures. The development of formal grammars allows mathematicians to leverage simply typed lambda calculus into the study of constituency grammar. This subsection of mathematical linguistics, known as categorial grammar, aims to study the functions responsible for combining constituents in a grammatical manner, *viz* with a view towards composition and compositional semantics. Theoretical work in this direction was motivated by theories such as AB-grammars [4, 10], Lambek calculus [76], Combinatory Categorial Grammar [124], Tree Adjoining Grammar [65], and Abstract Categorial Grammars [53].

Hindi has been largely underrepresented in these formal analyses of grammar, partly due to lack of exposure and partly due to the underlying strongly motivated syntactic analysis of the Paninian Grammatical Framework [17]. Therefore in this thesis, I aim to use a simplification of the Lambek calculus framework, known as pregroup grammar [78] in order to understand the syntactic phenomena underlying the language, when considered devoid of semantic considerations. Pregroup grammars are a simple categorial grammars which have been used to study languages like Japanese, Sanskrit and Persian, along with English, French, Hungarian and others, ensuring a diverse body of literature and surrounding operators and operations for representing constituency information.

In this thesis, I study the basic syntactic features of the Hindi language in terms of the word order, noun phrase characteristics, verb phrase characteristics, and formalize the agreement rules between them. I explore the representation of these properties in a consistent manner without relying on lexical semantic characteristics as much as possible. I then delve into the interoperability of relatively free word order syntactic constituent representations, their exploration in dependency syntax, and some pos-

sible applications of an advancement in constituency representation with the development of a hybrid grammaticality checker.

Contents

List of Figures

Figure Page

List of Tables

Chapter 1

Introduction

Categorial grammars are a group of formalisms that provide insight into constituents' arrangement in a given language. These formal representations of natural language syntax are classified as phrase structure grammars. One such formalism, namely pregroup calculus, has gained prominence due to its representation of free word order languages and applications in compositional distributional semantics. In this thesis, I use pregroup calculus to represent and study Hindi syntax's constituents and properties.

But why? Why is this a relevant problem? Why is the formal study of Hindi syntax essential? Why do I use pregroups as my syntactic formalism of choice? In this chapter, I take a step back to explore the motivation for this study, explain the relevance of this problem, and highlight the need to study the formal representation of natural syntax in the face of advances in contextualized neural language representations in advanced natural language processing.

1.1 Motivation

Syntax is the study of the structure of sentences in a given language and lies at the level between words and the meaning of utterances [31]. The study of phrase structure, word order, dependency relations, and the representation and generalization of sentence structure are all investigations in syntax. Some common theories of the study of syntax include dependency and categorial grammars, which are described here.

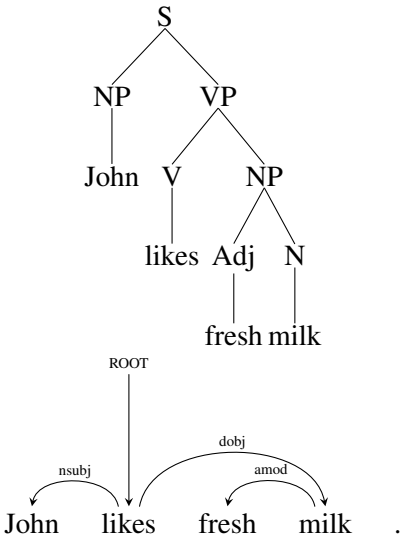
- **Phrase structure grammar**: Phrase structure grammar, also referred to as constituency grammar, is the study of syntactic constituents composed of contiguous words in a given sentence. Introduced by Noam Chomsky's surface structure of syntax, phrase structure grammar studies the relationship between constituents. It generalizes this relationship by decomposing the constituents that can grammatically occur around each other [95].

- **Dependency grammar**: Dependency grammar is the study of syntax based on identifying and classifying the relationship between the words of a sentence. The dependencies between words

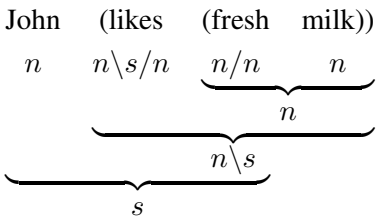
serve as directed links between words, which may be labeled based on the type of dependency between the two words [75].

- **Categorial grammar**: Categorial grammar is the study of syntax bound by constituent order in a given sentence and the rules for the composition for these constituents. These constituents are used to discuss the relationship between syntax and the underlying semantics defined by the words' lexical entries in the sentence [123].

Inherently, categorial grammars depend on constituency grammars, which identify and recursively type contiguous constituent elements until the sentence type has been reached, which is the root of the sentence. On the other hand, dependency parsing provides information about the relationship, optionally with labeled edges. We provide an example of a phrase structure (constituency parse) and a dependency parse of the sentence "John likes fresh milk".



The categorial grammar notation for the same sentence is provided below.



As can be seen, each of the above theories provides insight into the relationship between words and their categories, to different degrees of generalizability across languages and syntactic structures. Broadly, dependency grammars choose to type the relationships between the words of a sentence, while categorial grammars choose to type the words themselves. Because the latter relies on phrase and constituency structure, there is a general notion that *morphologically rich free word order languages are better represented using dependency grammars* [66].

Generally, languages with richer inflectional morphology are more likely to be free word order, as syntactic roles can be discerned from the inflections rather than the words' position. However, this implies that free word order languages do not have rules for the constituency and word order. The correctness of this generalization remains a pertinent question. We try to answer a specific instance of this question using Hindi syntax and a simple categorial grammar formalism, pregroups, adapted to handle word order alternation.

Hindi is a relatively free word order language, meaning that some word groups and phrases can be alternated with others. However, at an intra-chunk level, there is a rigid word order due to morphosyntactic properties of the language such as lexicalized case markers in the noun phrase and tense, aspect, and modality markers in the verb phrase. Hindi has a comprehensive dependency grammar due to its free word order nature, some of which can be deduced from the language's constituency structure. Due to these properties, Hindi syntax is the ideal case study for the formal representation of constituency for free word order languages, under some constraints.

1.2 Problem Statement

This thesis aims to study the representation and analysis of Hindi syntax under the computational algebraic constraints of the pregroup formalism. As mentioned in Section 1.1, Hindi acts as a case study for analyzing the constituency syntax of a free word order language. Given then extensive work in Hindi dependency grammar, I also use Hindi syntax to understand the nature and extent of categorial and dependency grammars' interoperability.

To that end, I answer three research questions about the syntax of relatively free word order languages:

1. How can the pregroup grammar formalism be adapted such that it can easily represent and parse restricted word order alternation?

2. How can the properties of Hindi syntax be represented within the pregroup framework?

3. What insights does the relationship between the dependency and categorial grammars can be generalized from this formal representation?

1.3 Why Pregroups?

In this section, I explain using pregroup as the grammar formalism of choice for the study of constituencies in Hindi. The formulation of the formalisms mentioned in this section is detailed in Chapter 3. This section intends to motivate the reason to use pregroup calculus.

Lambek calculus is a popular grammatical formalism used to describe phenomena in English syntax [76]. Also known as syntactic calculus, Lambek calculus aims to provide a significant test as to which a

given grouped string of symbols is a well-formed formula. Lambek calculus is a practical, comprehensive, and consistent representation of grammatical well-formedness based on reduction in a formulation [6]. In the example provided below, we parse the same sentence "John likes fresh milk" in the Lambek calculus framework.

The use of Lambek calculus is also distinguished from other categorial grammars in developing the notion of *'flexible constituency'*, the idea that constituents may overlap, allowing the words that are linked by dependencies to form a constituent [106]. This notion, known as dependency constituency, was one of the first notions of formalizing dependency grammars within a categorial framework [13]. However, the relative complexity of Lambek calculus, the lack of cut-elimination, and the syntactic rigidity due to the expressivity of the grammar belied its utility beyond the formal representation of English syntax only [92, 99].

Lambek calculus, much like its predecessor categorial grammars, has two function types, A/B, and $B\backslash A$. These function types and a non-commutative concatenation define the operations in Lambek calculus, using which several inclusion assertions are derived. However, this notation proved too rigorous, and due to the binary nature of the function types, it depended heavily on a rigid word order structure. In 1997, Lambek revisited the notion of categorial representations and introduced a simplified version of the original syntactic calculus, known as pregroups [78]. Pregroup calculus simplified the binary function types to *adjoints* (unary operators $-^l$ and $-^r$) as a simple notation of type representation. Therefore, pregroup calculus allowed simple concatenation rules, eliminating the need for the cut theorem, which allowed the representation of free word order in a simplified manner in this categorial framework.

Since its introduction, pregroups have been extended to numerous languages, such as French [87], Japanese [30], and Sanskrit [42] to name a few, while also exploring syntactic properties such as word order alternation [41], long-range dependencies [108], and feature agreement rules [34]. Pregroup calculus provides the extensibility in representing syntactic features, which is not afforded by most other lexical syntactic representations. Lastly, due to the basis of Lambek calculus in flexible constituencies and incorporation of dependency information, pregroups can be used to study the interoperability of syntactic dependencies and constituency grammar.

Pregroup calculus has also found extensive applications in natural language processing, in constructing an interpretable sentence representation for some implementations of compositional distributional semantics [49]. Compositional distributional semantics provides sentence embeddings, which are computed using vector representation of words and tensor products dictated by the compositional information provided by the categorial grammars. Pregroups can be well represented for such matrix-tensor operations due to their underlying category-theoretic structure.

1.4 Why Hindi Grammar?

In this section, I detail some properties of Hindi grammar that make it a prime candidate to be examined using a categorial syntactic formalism. Hindi is a highly inflectional language with a unique

word order paradigm dubbed *"restricted word order alternation"*. The syntax accounts for postpositions known as *karakas*, which align with the linguistic notion of case theory, as well as tense, aspect, and modality markers for verbs. All these functional markers are morpho-lexical, implying that some instances of these markers are inflections on the noun and verb phrase. In contrast, in others, they occur as individual words [68].

1.4.1 Karaka System and the Noun Phrase

Hindi's *karaka* system bridges the gap between case theory and thematic roles by providing context-dependent syntactic realizations of theta roles using monosyllabic markers for capturing the relationship between the noun and the verb. Secondary and tertiary postpositions use overloaded function words with *karaka* markers to provide semantic links such as locative, antessive, subessive, and causal information. Therefore, they play an important role in linking nominal and verbal constituents [74]. *karaka* markers are often overloaded to provide generally complementary syntactic information such as accusative/dative (*ko*) and instrumental/ablative (*se*). Hindi shows split ergativity with the null marker being used in the nominative case and the *ne* marker in the ergative case [71]. This extensive repository of functional markers that serve as both independent lexical units and inflectional morphemes highlights the relative importance of fixed word order **within** the constituents, which would then allow constituent shuffling rather than word order alternation [71].

The *karaka* system also expounds upon the question of language typology for Hindi. Generally, languages are described by one of two case combinations, **nominative-accusative** and **ergative-absolutive**. A nominative-accusative system is defined by the subject in the intransitive and transitive verbs are treated the same way, generally implying that there is no case marker for the subject. On the other hand, ergative-accusative systems are defined by the subject of the intransitive verb being treated the same way as the object of the transitive verb, *viz.* there is a case marker to denote the subject. English is a nominative-accusative language, while Basque demonstrates an ergative–absolutive case marking system. Hindi, on the other hand, demonstrates **split ergativity**. Hindi syntax has postpositions for the subject as well as the object (both accusative and dative objects), which can be seen in the examples below.

(1) rAm KaDA huA.
Ram.NOM stand get.PAST-PERF
'Ram stood'

As we can see in the example above, the subject does not have a case marker. The verb shows agreement with the subject in terms of gender and number.

(2) raam ne khaanaa khaayaa
Ram.ERG CASE food.ABS eat.PAST
'Ram ate food.'

(3) raam khaanaa khaataa hai
Ram.NOM food eat.HAB is
'Ram eats food.'

(4) raam khaane ko khaataa hai
Ram.NOM food.ACC CASE food eat.HAB is
'Ram eats food.'

(5) raam ne khaane ko khaayaa
Ram.ERG CASE food.ACC CASE eat.PAST
'Ram ate food.'

The above sentences show the relationship between the subject *Ram* and the object *food* under four different case marking methodologies. As we can see, the postpositions *ne* and *ko* are used to denote the subject and object respectively for some verb forms. Example (2) shows the ergative-absolutive form, (3) shows the nominative subject and an absolutive object with no case marking for either form, (4) shows nominative-accusative form, while (5) is the form that uses a postpositional marker for both the subject and the object [96].

1.4.2 Characteristics of the Verb Phrase

The Hindi verb phrase is marked by the presence of the main verb, an optional aspect marker, and a mood/tense marker. The aspect marker can be either inflected onto the main verb (such as *-taa* for the perfective aspect), or can be introduced as an independent lexeme (such as *rahaa* for the continuous aspect). Hindi uses an inflectional future tense marker, while the present and past tense markers are lexemes. Mood markers in Hindi are used *in place* of the tense markers; they do not co-occur. Other verbal inflections include the presence of two degrees of causativity, which are transitive in nature. The first causative is presented by using the suffix *-aataa* and the secondary using *-vaataa* [119].

Gender, number, and person agreement rules between the noun phrases and the verb phrase have multiple rules. The default gender is male and number in singular. The verb phrase agrees with the gender and number of the noun phrase which is "closest" to it, i.e. agreement depends on the use of postpositional markers.

Furthermore, Hindi provides an array of complex verb predicates, including verb-verb, noun-verb, and adjective-verb constructions. These complex predicates 'verbalize' a noun, and therefore, commonly occur with the verb *karnA* (do) and its various forms. However, it is not uncommon to see complex predicates with verbs such as *lenA* (take), *xenA* (give), *dAlnA* (put), and their various forms. Some examples of these constructions are provided below.

(6) raam ne khat **likh liyaa**
Ram.ERG CASE letter write do.PFV
Ram wrote a letter.

(7) nehaa ne mujhe **badhaaii dii**
Neha.ERG CASE to-me blessing give.PFV
Neha congratulated me.

(8) bachche hameshaa kaarpe.t **kharaab karte** hai.m
children.NOM always carpet dirty do.HAB is
Children always make the carpet dirty.

As we can see, individual verbs, or actions, are represented using complex predicates which include two components, the nominal, adjectival, or verbal"head word" which carries the semantic information of the predicate, and the verbalizer, which provides the predicate's syntactic information [93].

Hindi is generally considered a head-final language, where the syntactic head (which often corresponds to the semantically most important word) is the last word of a constituent. However, complex predicates (which have been typologically referred to as light verb constructions [26]) do not provide a head-dependent relationship and are a single semantic unit. Therefore, a providing lexical syntactic formalism for representing the complex predicates in Hindi verb phrases.

Verb phrases in Hindi pose a number of other challenges as well including infinitives, ditransitive verbs, and set of related verbs, including reflexives and indirect action forms. Verb phrases in Hindi pose a number of other challenges as well including infinitives, ditransitive verbs, and set of related verbs, including reflexives and indirect action forms [50]. Overall, the verb phrase as multiple morphological and lexical components which provide syntactic and semantic information about the verb of a sentence. The order in which these components occur is fixed, and therefore, it is possible to study the constituency grammar of the verb phrase and its relationship with the noun phrase in greater detail, with respect to agreement rules and long-range dependencies [69].

1.5 Applications

The study of categorial grammar representation of a language's syntax is a theoretical enquiry. Such an investigation aims to understand the relationship between the intuition of a "grammatical" sentence and an ideal framework representing the structural properties of sentences across languages. As I discuss in 1.3, pregroups have been adopted to a number of languages with a variety of syntactic properties. Due to this, they are capable of expressing constituency information in an easily parsable manner [16, 101]. Therefore, if we can describe the pregroup grammar of Hindi, it can be used to understand the grammaticality of sentences.

Therefore, with this exploration, I also develop insight into two potential applications of the pregroup representation of Hindi syntax: (a) grammar checking, and (b) interoperability of syntactic representations.

1.5.1 Grammar Checking

Grammar checking is the field of computational linguistics which aims to develop systems to detect whether a sentence is structurally sound, and isolate the cause of disfluency in the syntax if any. Generally, grammar checking systems are trained on large corpora of grammatically correct and incorrect sentences. However, with low resource languages like Hindi, we do not have access to a corpus of errors for training elaborate grammar checking systems. However, access to a system which is informed with the language's grammar trivializes the assessment of the grammaticality of a given sentence.

Universal dependency parsing extracts part-of-speech tags, head-dependent relationships, morphological features, and chunk level information. From this information, it is possible to infer the pregroup grammar, under some predefined language specific constraints known as metarules. Therefore, using universal dependency representation and parsing, it is possible to develop a representative pregroup grammar of the language (based on the training corpus of the dependency parser) and use that information to identify the grammaticality of an input sentence.

To that end, I develop a rule-based grammar checking module which uses dependency information and can serve as an add-on to neural dependency parsers. The grammar checker uses pregroup grammar as theoretical foundation and as it is rule-based, requires no pretraining. Such a module serves two purposes, namely: creating a hybrid, grammar-aware universal dependency parser with minimal pretraining and complexity, and determining sources of inaccuracy in the generated dependency parse due to inconsistency in the dependency parse and the pregroup reduction.

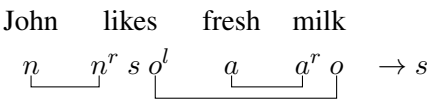
1.5.2 Partial Interoperability of Categorial and Dependency Parsing

As has been mentioned in section 1.2, one of the insights of developing a categorial grammar representation of Hindi syntax is the exploration of the relationship of the underlying constituent structure and its relationship with the dependency grammar.

Constituency grammar provides a context-free representation to the syntax of a language, which can be described in terms of terminal and non-terminal symbols. However, such a context free representation can not be constructed for the dependency grammar of a sentence, as the terminal as well as non-terminal symbols are words. Therefore, by definition, dependency parsing is a prediction task and does not *describe* the grammar of a given language.

As categorial grammars are established on the constituency framework of a language, the context-free representation is made more overt using a typing function for providing primary categories to words, and a reduction mechanism to generate the necessary parse structure.

As pregroups eliminate a number of the constraints of previous categorial grammar approaches, the parsing mechanism affords a diagrammatic representation of a sentence after it has been typed. For the sentence "John likes fresh milk" parsed in Section 1.1, the pregroup representation looks as follows.



In the diagram above, s is the type of the sentence, o is the direct object, n is the nominal subject, and a is the adjective. These types are akin to part-of-speech tags, and their arrangement with adjoints shows the relationship between a term and its surrounding words. $-^l$ and $-^r$ are left and right adjoints, which are named such that a left adjoint to the left a basic type and a right adjoint to the right of a basic type reduce. I cover the notion of typing a lexical item and the relationship between words and adjoints in detailed in chapter 3.

Diagrammatically, we can see that the dependency arcs and the reduction arcs from the pregroup calculus look very similar. The directedness of the dependency diagram can be established based on the adjoints as well. While this is a relatively simple example (in English), once equipped with word order alternation and long distance dependency functions, we see that this notion of partial interoperability between dependency relations and pregroup reductions becomes far more evident.

1.6 Summary of the Thesis

In this section, I summarize the rest of thesis and the contents of chapters.

In Chapter 2, I detail the relevant literature in the field of pregroup calculus and analysis. I explain the work done in various languages and extensions to pregroup calculus in order to represent various syntactic structures in a consistent manner. I also analyze trends in the evolution of literature for pregroup analysis, in comparison to other categorial grammars.

In Chapter 3, I provide the mathematical background of categorial grammars, pregroup calculus, Lambek's syntactic calculus and pregroup grammars. I explain the notations, the concept of typing, reduction, and provide formal mathematical basis of the concept used in word order alternation, *viz.* precyclicity. I also explain some important concepts in restricting word order alternation, such as cut elimination and the relationship between precyclic and non-precyclic elements for a pregroup grammar.

Chapter 4 explains the properties of the Hindi noun and verb phrase including the various postpositions, verb forms, modifiers, and syntactic markers. I introduce the basic types and their properties, including gender, number, person, and tense. I also showcase the examples of multiple sentences, their reduction, and highlight some syntactic properties.

Chapter 5 expounds upon some the representation and reduction of important syntactic criteria in the Hindi language, including restricted free word order, morphological type reduction, and long range dependencies. I use the type inventory detailed in Chapter 4 and further showcase how reductions occur outside the 'default' word order. Interestingly, these syntactic properties are not isolated to Hindi, which I show with examples from the English verb and preposition phrases.

Finally, in Chapter 6, I study that the relationship between categorial grammars and universal dependencies. I showcase the importance of language specific metarules in the extraction of universal dependency features, and conversion from these features into a pregroup parse. Using these phenomena, I explain the development of a minimal-pretraining rule based grammar checking module to complement universal depenency parsers.

Chapter 2

Literature Review

This chapter provides a comprehensive overview of literature for pregroup calculus, its operations and the various languages that have been represented in this formalism. Pregroup calculus has been further developed to include properties of precyclicity, tupled grammars, and additional operators, which allow it to represent a range of syntactic features and structures. I also explore the various languages represented by pregroups as a framework, and detail the parsing mechanisms developed for them. Finally, I showcase some work done in the applications of pregroup calculus which are significantly different from other categorial grammars, i.e. mathematical foundations for compositional distributional models of meaning.

2.1 Categorial Grammars

Categorial grammar encompasses a number of related formalisms which associates elements of phrase structure language to categories and defines functions to represent syntactic or semantic compatibility and compositionality. Categorial grammars evolved from the works of Ferge (*Begriffsschrift*) [60], which aimed to determine the truth value of a statement using first order predicate logic. The development of this combination of functions and arguments has led to the development of lambda (λ-) calculus and its variants, the basis of programming language syntax. Early works by Ajdukiewicz [4], Bar-Hillel [10, 11], and Lambek [76] provided the first basis of a notation which aimed to investigate constituency-based syntactic phenomena in natural language derived from typed λ-calculus [123].

Adjukiweicz's and Bar-Hillel's contributions (known as AB-calculus [91]) to capturing and representing syntactic notions laid the foundation for the modern notation, description, and nomenclature in categorial grammars [91]. Generally, these grammars represent syntactic types primitive and complex categories. Primitive categories are *usually* limited to S (sentence), N (common nouns), and NP (noun phrase). Complex categories are formed by operations between primitive categories [58]. Since the development of Lambek calculus, most categorial grammars use the schema of application known as *Functional Application Rule Schema* [7, 32]. Categorial grammars also aim to establish the relationship

between syntactic types and semantic categories (specifically Husserl's Meaning Categories [64]) using operations that represent compositionality in constituency grammar.

Many theories of categorial representation evolved from AB-calculus, which include representations such as Tree-adjoining Grammars (TAG) [65], Link Grammars [117], Combinatory Categorial Grammars (CCG) [124], and Abstract Categorial Grammars (ACG) [53]. TAGs were among the first formal representations which generated 'mildly context sensitive' grammars, and the extensions to the grammar included type substitution as a combining operation in their reduction mechanism. TAGs also showcased 'reduction' for extracting parse information, which became the basis for deriving weak equivalence between tree-adjoining grammars and combinatory categorial grammars [126].

On the other hand, Lambek's syntactic calculus also established the first adjoint and complement notion using directional reduction and combination as operations for representing syntactic properties in contiguous constituents [76]. Directional reduction led to the development of the notions of left and right adjoints, type raising, composition, and the derivation to equivalencies to Curry's semantic calculus [39]. Syntactic calculus was the first categorial grammar to establish a formal notion of primitives which resembles syntactic categories which account for inflections such as gender, number, tense, and aspect. Syntactic calculus also represented a much larger set of primitives which include statement and question (which were types of the main verb), modifiers, nouns and pronouns, auxiliary verbs, and prepositions.

Theoretically, syntax refers to an adjunct as an 'optional element' of the constituent while a complement is a 'mandatory' or 'obligatory' element. In lexical semantics, adjuncts modify the head word, while complements complete the meaning of the head word [58]. For example, in the constituent: '*a red apple*', '*a*' is a necessary part of the phrase, so it is considered a complement, while '*red*' is an optional modifier and descriptor, therefore it acts as an adjunct. Categorial grammars derive the relationship between the the head and the rest of the constituents using structures known as *head-complement* and *head-adjunct* structures.

2.2 Pregroup Calculus and Pregroup Grammar

In 1997, Lambek simplified their syntactic calculus by eliminating the cut operations. This change in representation alienates pregroups from other categorial representations which use syntactic calculus' notion of cut operations for determining adjunct and complement directionality. The eliminated cut operations are replaced with basic types augmented with adjoints [78, 79]. Lambek introduces a left and right adjoints which interact with the other basic types using a single compositionality operator, which allows the visualization of language in terms of valencies (where the head demands or needs a given adjunct or complement) [82]. These modifications eliminate the difference between a head and a complement, therefore removing the distinction between a head-complement and head-adjoint structures [19, 22]. By combining the notion of complement and adjunct, pregroup calculus analyzes the relationship between constituency heads and their adjoints as a system of valency, which is also used by dependency grammar without the barriers of constituency.

While pregroup calculus represents the relationship between basic types, pregroup grammar introduces a type-logical context-free grammar [19]. Pregroup grammar elaborates the relationship between words (tokens) and their types, as well as formalizes the process of reduction. The logic used to define the reationsip between the types and their adjoints is known as compact bilinear logic and is fundamental in the repreentation of syntactic phenomena such as free and restricted word order movement as well as long distance dependencies [25]. Other advances in pregroup grammar include tupled pregroup grammar, which details the relationship between different adjoints under the constraints of proper typing [122]. Finally, properties of the generalizations of pregroup grammar based on their sequences was studied as 'substructural logics', closely associated with compact closed categories [23]. This analysis was crucial in determining the equivalence between sequences represented by Lambek's original syntactic calculus, and the relatively modern pregroup notation [24].

2.3 Pregroup Grammars of Various Languages

Pregroup calculus and the surrounding framework was introduced as a representation for English syntax. However, soon after its introduction, the calculus was used to represent and analyze syntactic structures in a number of other languages. These extensions into other languages was done either by:

- exploring a specific syntactic feature or phenomena (such as free word order, causative constructions, and long distance dependencies), or
- elaborating the representation of syntactic constructions in a given language (such as statements and questions in Italian, French, etc.).

The former provides insight into the mathematical and computational insights and properties required in order to represent phenomena which might be useful to the representation of other languages as well. Unlike most of the other categorial grammars, there has been a lot of work done for accommodating free word order languages into the pregroup framework, including clitics in Italian [37], word order for Hungarian [114] and Sanskrit [42], and the introduction of cyclic pregoups to generally develop a generalized framework for this property [40, 43]. Pregroup calculus has also been used to represent morphosyntactic features such as causative verb constructions in Japanese [30], long distance and feature agreement in French [108, 87].

On the other hand, studies in the generalization of pregroup representation across languages have been done for a plethora of languages. Properties and representation of English syntax have been explored in stages from simple statements [82], questions [83], imperatives [88], relative clauses [86], and nominal and adjectival phrases [84]. A complete treatise of pregroup representation of English sentences was detailed by Lambek [85].

Beyond English, pregroups have been extensively for processing French, including noun phrase [56, 57], verb phrase [112], clitics [55] as well as computing agreement rules in efficient manner [87]. The

work on pregroups in French led to the development of 'semantic pregroups' which detailed subtyping to handle relationships between basic types. Italian has also been very well represented using the pregroup framework, with statements and questions [33], clitics [38, 35, 36], and agreement rules [34] being extensively studied.

Languages like German [80, 89, 90], Hungarian [114], Persian [113], and Sanskrit [42] have been represented for their word order. While pregroup calculus in German and French was intended to handle basic typing due to word order information, for Sanskrit and Hungarian, cyclic and precyclic pregroups were introduced to represent free word order syntax in a uniform representation [41].

On the other hand, languages like Chinese [12] and Japanese [28, 29] for their morphosyntactic properties, such as computing agreement based on gender, number, person, honorifics, case, and tense. Based on the cross-lingual study of pregroup representation of syntactic and morphosyntactic phenomena, a complete thesis and guidelines for the adaptation of pregroups was established by Kilask et. al. [72].

In both these modes of exploring syntactic phenomena across various languages, assigning basic types is a fundamental step, as it dictates the symbols used by the calculus and their relationship with the syntactic types of the language [81]. While the basic types for English were initially assigned from the adaptation of most categorial grammars, i.e. the noun, the noun phrase, and the sentence, other basic types for other parts of speech were introduced as well [85]. These basic types were introduced in order to type and compute sentences presented by other examples in the study of English syntax [86]. With the introduction of basic types to other languages, two methods have been adopted in literature. For languages like French and Italian where the noun phrase, the verb phrase, clitics, and interrogative sentences have been studied in detail, basic types were introduced based on examples in existing literature. For example, the French noun phrase introduces different basic types for mass nouns and count nouns [57], which is introduced for efficient computations and parsing. On the other hand, literature which explores the representation of a particular syntactic phenomena has adopted the basic type set from English (such as word order in Hungarian and Sanskrit), or an appended set to include the part-of-speech being studied (for example causatives in Japanese). A uniform methodology for the establishment of a given set of basic types has not yet been established in literature, unlike in other categorial grammars.

2.4 Pregroup Parsers

In order to harness the representative power of pregroups, syntactic parsers were introduced for the corresponding pregroup grammar of English. Pregroup grammars are weakly equivalent to context-free grammars [25]. Therefore, pregroup parsers follow the same methodology of parsing context free grammars, more specifically, a group of languages called Savateev (Free Compact 2-Category) languages [111]. Algorithms for polynomial time and linear time parsing of simple pregroup grammars have been introduced for a class of pregroup parsers [115]. Since pregroup grammars are cateogrial, and therefore

a representation of compositional semantics, some parsers use majority or partial composition which leverages syntactic patterns for efficient processing [116].

Pregroup parsers are based on the idea of partial and majority composition. Initial algorithms for complete and parsing of pregroup grammars were of cubic time complexity, which are adaptations of context free parsing algorithms such as Early's and CYK [15]. However, these algorithms are not efficient as they do not exploit any of the constraints of the pregoup calculus itself. Two schools of pregroup parsing introduced numerous algorithms which generalized pregroups to different mathematical objects [107, 101]. One school provided dynamic processing of pregroups in polynomial time based on the idea that in representing language, parsers always reduce the length of the string [100, 102]. On the other hand, polynomial time, and subsequently linear time algorithms, were developed under the generalizations from categorial grammars based on patterns in constituent analysis and the relationship between basic types [14, 16].

Advances in pregroup parsing have also enabled the pregroup representation of discourse structure information [109]. However, discourse structure representation is a mildly context sensitive notion, which have also been explored in the pregroup parsing literature with polynomial time algorithms [110].

2.5 Applications in Compositional Distributional Semantics

Pregroup calculus was introduced as a categorial representation of numerous syntactic properties based on the group- and category-theoretic notions. Consequentially, pregroup calculus was used to characterize compositional semantic notions when developing the mathematical foundations for compositional distributional models of meaning [46, 49]. The aim of compositional distributional semantics is to mathematically model the meaning of a sentence using the vector representation for individual words, and categorial grammar to analyze the relationships between them [46]. As mentioned before, pregroup calculus is a simplified categorial grammar which maintains the expressivity of complex calculi. The group-like mathematical properties allow smooth interoperability between vector calculus and category theory, using tensor calculus [127]. The theory was introduced as categorical compositional distrbutional semantics [70].

Since its development, the cateogry-theoretic tensorial transformations of pregroup calculus has used to explain syntactic phenomena which are not purely compositional in nature, such as the pet fish phenomenon [48], literal and metaphorical sense analysis [62], and a generalized quantifier theory [63]. The theory of compositional distributionality has also been used to analyze lexical semantic phenomena such as graded entailment [8] and hyponymy [9, 51]. Furthermore, compositional distributional models based on pregroup calculus provide insight into event semantics as well, which is useful in sequentiality and cause analysis [105]. Pregroups have also been shown to have equivalent compositional representative expressivity to Lambek's syntactic calculus [47].

2.6 Conclusion

This chapter introduced and detailed the literature that led to the development of categorial grammars such as pregorup calculus for the representation and parsing of syntactic structures. A brief history of categorial grammars was provided to elucidate the context for the development of pregroups. In this chapter, I highlight how pregroup calculus was developed for representing phenomena in English syntax, and how it has become a popular choice for diagrammatizing syntactic structures across languages and language families. An overview of the parsing tools and methodologies for pregroup grammars was also introduced, and finally, the application of pregroups in mathematical models of compositional distributional semantics was elucidated.

Chapter 3

Categorial Grammars, Pregroups, and Precyclicity

In this chapter, I introduce the mathematical foundations of pregroup calculus and pregroup grammars, including the definitions of a monoid, a group, partial order, and categorial grammars. I then go on to explain the definition of pregroups based on the foundations, study their operations and some of the lemmas associated with them. Furthermore, I explore pregroup grammars based on this calculus, and elaborate on precyclicity as an important construct for representing word order in natural languages. Finally, I showcase the equivalence in operation between precyclic and non-precyclic pregroups, which is very useful for languages with restricted word order movement.

3.1 Mathematical Background

Pregroups are mathematical objects which define the relationship between the elements in their set. In order to understand pregroups and pregroup calculus, we first start by defining the most elementary notion of operations within a set, a monoid.

Definition 3.1.1 (Monoid)**.** A set equipped with a binary operation $S \times S \to S$, which is denoted by $\cdot$, is a monoid if it an identity element is a member of the set, and the operation is associative.

Associativity: $\forall a, b, c \in S : (a \cdot b) \cdot c = a \cdot (b \cdot c)$

Identity Element: $e \in S$ such that $\forall a \in S : e \cdot a = a \cdot e = a$

There is only one identity element in the a given monoid, which can be considered a constant. Therefore monoids can be defined by the triple $(S, \cdot, e)$. An example of monoids include natural numbers under addition (where the identity element is 0), and under multiplication (where the identity element is 1). Another example is the set of all subsets: given a set S, the set of all subsets of S (known as the power set S_P is a monoid under union (with the identity element being ϕ) and intersection (with the identity element being S itself).

Another important concept in the study of pregroups is the notion of partial order.

Definition 3.1.2 (Partial Order)**.** A partial order is a homogeneous binary relation $\leq$ over a set P which satisfies three axioms: reflexivity, antisymmetry, and transitivity. When $a \leq b$, it can be said that a is related to b, or a reduces b.

Reflexivity: Every element in a partially ordered set is related to (reduces) itself $\forall a \in P : a \leq a$

Antisymmetry: For $a, b \in P$: $a \leq b$ and $b \leq a$, then $a = b$.

Transitivity: For $a, b, c \in P$: $a \leq b$ and $b \leq c$, then $a \leq c$.

The most common example of a set with partial order (also known as a *poset*) is real numbers ordered by the less-than-or-equal-to relation. The power set S_P of a given set S is a poset if ordered by the inclusion relation. Similarly, the set of all strings ordered by substrings are also posets.

3.2 Pregroup Calculus and Pregroup Grammar

With the foundations of a monoid and partial order, I can now define a pregroup.

Definition 3.2.1 (Pregroup)**.** A pregroup is a partially ordered monoid adorned with two unary operators, the left adjoint (denoted by $-^l$) and right adjoint (denoted by $-^r$), which have the following property.

$$x^l \cdot x \leq 1 \leq x \cdot x^r$$

Pregroups are defined by $(P, 1, \cdot, -^l, -^r, \leq)$, where P is the set, 1 is the identity element, the $\cdot$ operator is a concatenation operator (usually not explicitly mentioned) and the $\leq$ operator indicates partial order (which is sometimes written as $\rightarrow$).

Pregroups are based on the relationship between products of basic types and their adjoints. Some properties of pregroups include:

$$1^l = 1 = 1^r \tag{3.1}$$
$$(a \cdot b)^l = b^l \cdot a^l \tag{3.2}$$
$$(a \cdot b)^r = b^r \cdot a^r \tag{3.3}$$
$$(a^l)^l = a^{ll} \tag{3.4}$$
$$(a^r)^r = a^{rr} \tag{3.5}$$
$$a^{rl} = a = a^{lr} \tag{3.6}$$

Adjoints are switching in nature, which means that:

$$a \rightarrow b \implies b^l \rightarrow a^l \tag{3.7}$$
$$a \rightarrow b \implies b^r \rightarrow a^r \tag{3.8}$$

Pregroup calculus defines a basic type as an element $a \in P$. A simple type can be obtained by basic types as:

$$a^{lll}, a^{ll}, a^{l}, a, a^{r}, a^{rr}, a^{rrr}$$

A compound type is then defined as a concatenation of simple types using the $\cdot$ operator. One method of representation of pregroups assumes numerical exponentiation of adjoints, i.e. $(-)^2$ as opposed to labeled $(-)^l l$. In order to use them interchangably, we assume that positive exponents are left adjoints, negative exponents are right adjoints, and the absolute value refers to the number of adjoint operations done on the type. Therefore $(-)^{lll}$ is equivalent to $(-)^3$. All the properties of free pregroups can be derived from first principles. Hence, for the purpose of representing syntactic characteristics, the terms *free pregroup* and *pregroup* are treated equivalently.

We can now redefine contraction and expansion using this method, as well as present derivation of induced steps. For $p, q \in P$ in $(P, \leq)$, the basic rewriting rules are:

$$Xp^n p^{n+1} Y \to XY \quad \text{(contraction, CON)} \tag{3.9}$$

$$XY \to Xp^n p^{n+1} Y \quad \text{(expansion, EXP)} \tag{3.10}$$

$$Xp^{2n}Y \to Xq^{2n}Y \text{ and } Xq^{2n+1}Y \to Xp^{2n+1}Y, \text{ if } p \leq q \quad \text{induced steps (IND)} \tag{3.11}$$

In order to use pregroups to represent natural language, certain rules regarding the behaviour of iterating adjoints such as $(-)^{ll}$ and $(-)^{rr}$ must be established. This implies the existence of a generalized rule for contraction (where the partial order results in an expression with fewer elements) and for expansion (where the partial order results results in more elements of the pregroup being introduced). To that end, we define a free pregroup [59].

Definition 3.2.2 (Free Pregroup)**.** Let $(P, \leq)$ be a partially ordered set of primitive types. Integer exponents on P entail:

- $P^{(Z)} = \{p^{(i)} \mid p \in P, i \in \mathbb{Z}\}$: the set of atomic types,
- $Cat_{(P,\leq)} = (P^{(Z)})* = \{p_1^{(i_1)} \dots p_n^{(i_n)} \mid 0 \leq k \leq n, p_k \in P, i_k \in \mathbb{Z}\}$: the set of all types in the calculus,
- $\leq$ is the smallest partial order relation which obeys for $(p, q \in P)$, $(X, Y \in Cat_{(P,\leq)})$:
  - $Xp^{(n)}p^{(n+1)}Y = XY$: generalized contraction (GCON)
  - $XY = Xp^{(n+1)}p^{(n)}Y$: generalized expansion (GEXP)

Note that GCON is the application of IND on CON, and GEXP is the application of IND on EXP for two elements $x, y \in P$. Free pregroups also have the property that without loss of generality, contractions precede expansions. This is explained as follows[79]

Definition 3.2.3 (Switching Lemma)**.** If $X \leq Y \in (P, \leq)$, there exists a string $a \in Cat_{(P,\leq)}$ such that $X \leq a$ by generalized contractions (GCON) only and $x \leq Y$ by induced steps (IND) only.

Given that the calculus has been established in detail, I can now establish the mechanism used to syntactically type and reduce sentences. This procedure requires strings in a given language, the sequents generated by the pregroups, and a notion of type assignment. This mechanism is explored in the form of a pregroup grammar, a categorial grammar which leverages pregroup calculus for type analysis and reduction.

Definition 3.2.4 (Categorial Grammar)**.** A cateogorial grammar is defined as a quintuple $G = (\Sigma, I, S, P(T), T)$ where Σ is the alphabet (often represented by the words in the sentence being parsed); $P(T)$ is the set of primitive generated from a set of basic types T; a main type $S \in P(T)$ associated with the sentences, and a function $I : \Sigma \mapsto P(T)$ which maps each element of Σ to a type in the set of types $P(T)$.

A sentence is said to belong to G if it can be assigned types from $P(T)$ such that they derive S according to the rules of the type calculus. Pregroup grammars are specialized categorial grammars defined below [19].

Definition 3.2.5 (Pregroup Grammar)**.** A pregroup grammar is a quintuple $G = (\Sigma, P, \leq, s, I)$ such that Σ is a nonempty, finite alphabet, $(P, \leq)$ is a finite poset, $s \in P$ is the type of the sentence, and I is a finite relation between symbols from Σ and non-empty types (on P).

Contemporary literature symbolizes the set of all basic types as $\mathcal{B}$, and the set of all compound types as $T(\mathcal{B})$. $\mathcal{B}$ is a partially ordered set, while $T(\mathcal{B})$ is a free, proper pregroup over the set $\mathcal{B}$.

3.3 Lambek's Syntactic Calculus and Pregroup Grammar

Lamebek's pregroup calculus was introduced as a simplification of their previous work on syntactic structures and categorial grammar. The construction, known as Lambek calculus or syntactic calculus. In this section, we explore the relationship between the two calculi.

The building block of Lambek's syntactic calculus is a residuated monoid, analogous to a pregroup in pregroup calculus [77].

Definition 3.3.1 (Residuated Monoid)**.** A residuated monoid is a structure $\mathcal{G} = (G, \leq, \cdot, /, \setminus, 1)$ where $(G, \cdot, \leq)$ is a partially ordered monoid, and $/$ and $\setminus$ are binary operators under which iff $ab \leq c$, iff $a \leq c/b$ and iff $b \leq c \setminus a$, $\forall a, b, c \in G$.

The object presented above is known as a residuated monoid as the set G forms both a poset $(G, \leq)$ and a monoid $(G, \cdot, 1)$, such that monotony conditions of residuation (the binary operators and their

directionality) can be derived by inverting the operations. Note that $\cdot$ is not commutative, which is why two binary operations of left and right reduction are postulated.

Given the definition, we see that residuated monoids are quite similar to pregroups in structure. The only difference between a residuated monoid and a pregroup is the use of binary left and right reduction operations, as opposed to the unary left and right adjoints. However, the differences in the operators makes pregroups easier to represent, as basic types with adjoints are treated the same way as basic types without adjoints, except when they are of the same base type. More importantly, in any pregroup that can be defined, the equivalency $a \setminus b = a^r b$ and $a/b = ab^l$ hold. Therefore, every pregroup can be expanded to a residuated monoid. This implies that a given sequent in Lambek calculus with an 'empty antecedent derivation' is valid in all pregroups [21]. However, the converse is not true. $a \cdot (b/c) = (a \cdot b)/c$ holds true in every pregroup, as both can be represented as $a \cdot b \cdot c^l$, but this is not true for every residuated monoid.

3.4 Cut Elimination and Precyclicity

In this section, I explain the cut theorem, the cut elimination of pregroup calculus, and precyclicity. For this purpose, it is useful to picture a string produced by a given calculus as a list of derivations which can be written in the form of sequents, i.e. a list of conditional tautologies [52]. In doing so, I present the derivation of cut elimination in a manner easier to intuit.

Cut theorem is an essential part of proofs in sequent calculus, and therefore by extension, all of formal logic. However, there are some languages in which any proof made with cut theorem can also be made without using its derivation. These languages are named **cut eliminated**. I provide in the subsection below the cut theorem in Lambek's syntactic calculus, and the intuition towards deriving cut elimination in pregroup calculus.

One of the consequences of cut elimination is the derivation of the relationship between genenral and precyclic pregroups. General pregroup calculus does not include any notion of commutativity or cyclic properties. However, some pregorups can include a lose form of cyclic interaction known as precyclicity. Precyclic pregroups perform derivations the same way as general pregroups, with precyclic derivations known as transformations. Using the aforementioned cut elimination of pregroups, we prove that a type in a general pregroup is not affected by transformations, and can therefore be reduced by a type in a precyclic pregroup given the appropriate reduction rules.

3.4.1 Cut Theorem and Cut Elminiation of Pregroups

In order to define the cut theorem, I first rewrite the known rules of pregroup calculus using a rewriting system for sequent analysis. The $(P, \leq)$ is fixed, and atoms and types are those defined above.

$$\mathrm{X} \implies \mathrm{X} \text{ (ID)}$$

$$\frac{X, Y \implies Z}{X, p^{(n)}, p^{(n+1)}, Y \implies Z} \text{ (CON)}$$

$$\frac{X, Y \implies Z}{X, Y \implies Y, p^{(n+1)}, p^{(n)}, Z} \text{ (EXP)}$$

$$\frac{X, p^n, Y \implies Z}{X, q^n, Y \implies Z} \text{ (LIND)}$$

$$\frac{X \implies Y, p^n, Z}{X \implies Y, q^n, Z} \text{ (RIND)}$$

For this rewriting, the rule of induced steps (IND) is consrtucted by splitting the rule into left and right induced steps (LIND, RIND respectively). As before, LIND and RIND are constrained by $p \leq q$ if n is even, and $1 \leq p$ if n is odd.

Given these rules, the cut theorem is defined as:

$$\frac{X \implies Y \qquad Y \implies Z}{X \implies Z} \text{ (CUT)}$$

In these derivations, two notions of reduction are presented, one represented by the bar in $\frac{A \implies B}{C}$, which is equivalent to $\rightarrow$ in pregroups. The other, $\implies$, is the reflexive and transitive closure over $\rightarrow$. Clearly, a system of calculus enriched with the cut theorem can create reductions in more sequents. The relationship between $\implies$ and $\rightarrow$ makes cut theorem very important, however, note that in pregroup calculus, a transitive closure implication can be derived entirely using generalized contractions, and consequently, the switching lemma.

Therefore, it can be strictly proven that any sequent derivable using cut theorem in pregroup calculus can be derived without it due to the switching nature of adjoints. Further, any and all valid proofs in pregroup calculus can be proved without using the cut theorem if and only if they can also be proved using it [20, 98]. Hence, pregroups are said to be cut-eliminated.

3.4.2 Precyclicity

In this section, I introduce precyclicity, the notion required to represent free word order in syntax. I also explain the relationship between precyclic and non-precyclic pregroups using the aforementioned cut-elimination of pregroups.

A detailed understanding of cyclic properties of a pregroup can be derived from Lambek's syntactic calculus, and using a translation between residuated monoids (the structure used in syntactic calculus) and pregroups [39]. Since pregroups used for language formalism are free, proper pregroups [19], the classical definition of cyclicity $a \cdot b \leq c \implies b \cdot a \leq c$ does not hold. However, a weak form of cyclicity, called precyclicity, is admitted, which has the following properties [128]:

$$pq \leq r \implies q \leq p^r r \tag{3.12}$$
$$q \leq rp \implies qp^r \leq r \tag{3.13}$$
$$pq \leq r \implies q \leq rp^l \tag{3.14}$$
$$q \leq pr \implies p^l q \leq r \tag{3.15}$$

Due to this, we obtain the following rules for precyclicity with double adjoints [1]:

$$1 \leq ab \xrightarrow{ll} 1 \leq ba^{ll} \tag{3.16}$$
$$1 \leq ab \xrightarrow{rr} 1 \leq b^{rr}a \tag{3.17}$$

Here, ba^{ll} and $b^{rr}a$ are know as the precyclic permutations of ab. Given these precyclic permutations, for $A, B, C \in P$, the following precyclic transformations are defined:

$$(ll)\text{-transformation: } A \leq B(ab)C \rightsquigarrow^{ll} A \leq B(ba^{ll})C \tag{3.18}$$
$$(rr)\text{-transformation: } A \leq B(ab)C \rightsquigarrow^{rr} A \leq B(b^{rr}a)C \tag{3.19}$$

These precyclic transformations provide the following two equations,

$$p^r q \leq qp^l \tag{3.20}$$
$$qp^l \leq p^r q \tag{3.21}$$

which can be seen to be empirically derived in clitic movement patterns in other languages, explored in [40]. The operations between the sets $T(\mathcal{B}_T)$ and $T(\mathcal{B}_{NT})$ that were implemented in the paper in Section 5.2 and the non-precyclic nature of $T(\mathcal{B})$ are explained here.

Precyclic transformation was introduced in [128], for relaxing the conditions on the cut theorem on one sided sequent calculus for pure non-commutative classical linear propositional logic. We define a grammar G wherein elements $\Gamma, \Delta \in G$ and $A \in \text{seq}(G)$, where $\text{seq}(G)$ is a sequent in G.

The original rules of precyclic transformation in sequent calculus were written as:

$$\frac{\Gamma \implies A}{A^{\perp\perp} \implies \Gamma}$$

$$\frac{A \implies \Gamma}{\Gamma \implies {}^{\perp\perp}A}$$

Clearly, the position of the $\perp$ symbolizes the direction of the adjoint relation, and the number of $\perp$s determines the degree of the adjoint. The calculus for which this rule was applicable was called SPNCL', while the calculus where this rule was not applicable was SPNCL. In order to understand the affect of this theorem, note the cut theorem in SPNCL:

$$\frac{\Gamma_1 A \implies \Gamma_2 \qquad \Delta_1 A^{\perp} \implies \Delta_2}{\Delta_1\Gamma_1 \implies \Delta_2\Gamma_2}$$

$$\frac{\Gamma_1 {}^{\perp}A \implies \Gamma_2 \qquad \Delta_1 A \implies \Delta_2}{\Delta_1\Gamma_1 \implies \Delta_2\Gamma 2}$$

if $\Delta_1 = \varnothing$ or $\Gamma_2 = \varnothing$.

And the cut theorem in SPNCL', due to these rules, was reduced to:

$$\frac{\Gamma \implies A \qquad \Delta \implies A^{\perp}}{\Delta \implies \Gamma}$$

$$\frac{\Gamma \implies {}^{\perp}A \qquad \Delta \implies A}{\Delta \implies \Gamma}$$

Precyclic pregroups are a result of a bilinear mapping of this reduction in SPNCL' to pregroups, using the interpretation map of compact bilinear logic [21]. However, note that this reduction was made to relax the constraints on cut theorem in SPNCL. [20] notes that pregroups are cut eliminated, which means that all properties that can be proved using cut theorem can be proved without it. Due to this cut-elimination, in a pregroup mapped from SPNCL and one mapped from SPNCL', the adjoint behaviour is identical and therefore their concatenation and reduction do not undergo any change.

Now, the non-precyclic nature of the pregroup $T(\mathcal{B})$ is to be proved. $T(\mathcal{B})$ has been defined as $T(\mathcal{B}_T) \cup T(\mathcal{B}_{NT})$, where $T(\mathcal{B}_T)$ is the pregroup that allows for precylic transformations and $T(\mathcal{B}_{NT})$ is the pregroup that does not allow the same. Note that both SPNCL and SPNCL' are defined over the same set of sequents, but are defined using different formulae. For a formula A of $\mathscr{L}$(SPNCL) and formula B of $\mathscr{L}$(SPNCL'), the cut theorem in SPNCL' will not be applicable for a proof with both A and B, as the rule $(-)^{\perp\perp}$ can not be used for formulae of SPNCL. Hence, given the compact map from bi-linear logic to pregroups, the analogous transformations become inapplicable over a pregroup which has *any element* that does not obey precyclicity. Therefore, the union of a precyclic and a non-precyclic pregroup is a non-precyclic pregroup.

3.5 Conclusion

In this chapter, I explained the mathematical basis of pregroup calculus and pregroup grammar. First, I define and explain the algebra relevant to the definitions and properties of pregroups, including monoids and partial order. I then go on to define a pregroup, its operations and properties, and the notion of a pregroup grammar. I further elaborate on a representation mechanism of generalized contractions and expansions, and provide insight into the cut elimination of pregroups, an important facet in understanding precyclicity and the relationship between precyclic and non-precyclic pregroups.

Chapter 4

Representing Syntactic Constituents

In this chapter, I explore the various constituent groups in Hindi syntax, including the noun phrase, which includes nouns, pronouns, demonstratives, case markers and nominal modifiers, and the verb phrase, which include verbs, verbal auxiliaries (aspect and tense-mood markers), verbal modifiers, causatives, and infinitives.

4.1 Pregroups and Natural Language Syntax

In this section, we provide an overview of basic and compound types, typing rules, reduction, and using pregroups as in order to derive the syntactic structure of sentences. For example's purposes, I first illustrate the analysis on English, based on Lambek's works [78, 79, 83].

In type-checking sentences, the final type of the sentence is conventionally the same as the type of the main verb, e.g. s for statements. For example, in English a simple transitive verb will be represented as follows:

subject verb object

$$\underbrace{n}\quad \underbrace{n^r\ s\ o^l}\quad \underbrace{o}\quad \to s$$

, where n is the type of the nominal subject, o is the type of the object, and s is the type of the main verb. The reductions are shown by the arcs, and the sentence reduces to type s.

Each arc signifies a reduction, wherein the full derivation would be as follows:

$$\begin{aligned}
&(n) \cdot (n^r s o^l) \cdot (o) \\
&\to (n \cdot n^r)\ s\ (o^l \cdot o) && \text{(changing brackets)} \\
&\to 1 \cdot s \cdot 1 && (a \cdot a^r = 1 \text{ and } a^l \cdot a = 1) \\
&\to s && (1 \cdot a = a \text{ and } a \cdot 1 = a)
\end{aligned}$$

Lambek defines a list of basic types for English syntax [89]. These basic types were determined based on the parts-of-speech, syntactic roles, morphological characteristics such as gender, number, and person, as well as sentence type (such as imperative or interrogative).

In this chapter, I define the basic types for Hindi in a very similar manner. Each sentence in Hindi comprises at least one noun phrase and one verb phrase. Each noun phrase may contain nouns or pronouns (which may be the subject, direct object, or indirect object in the sentence), along with a postpositional marker. This marker (*vibhakti*) represents the relationship between the noun and the verb (*karaka*), or the noun and another noun phrase (*sambandha*). A noun phrase may also contain nomnial modifiers. A verb phrase consists of the verb, in some cases the verbalizer (i.e. the word which converts a noun into a verb), and a tense marker, as well as some modifiers. Verb phrases also have rich morphological inflections for causativity and aspect marking.

Unlike most grammars, the pregroup grammar of Hindi is not strictly lexicalized in order to maintain some uniformity of representation. Markers for the same syntactic feature can be either independent or dependent morphemes in Hindi, which leads to complexities in strictly lexicalized. This is detailed further in chapter 5.

4.2 Noun Phrase

In this section, I introduce the noun phrase in Hindi. The noun phrase can refer to the subject, object, or any other thematic role which is provided in the sentence. These phrases consist of either a noun or a pronoun. The noun phrase has features of gender, number, and person, although gender is a latent feature and is not generally represented in the noun using a morpheme. The noun phrase may be followed by a marker which signifies its role in the sentence with respect to the verb it is associated with (analogous to case marking). This relationship is characterized by a lexical marker known as a *vibhakti*, and the relationship it characterizes is known as a *karaka*. A noun phrase may also be related to another noun phrase via the possessive case. This is represented in Hindi using a *sambandha* marker.

4.2.1 Pronouns in Hindi

(9) tum -ne
You ERG
'You' (in ergative case)
$(\pi_{12}\ \underbrace{\kappa_1{}^l)(\kappa_1}) \rightarrow \pi_{12}$

Pronouns in Hindi are inflected by number, person and case, except for in second person, where the singular and plural word form remains the same. Case inflections mean that the *vibhakti* of the respective role is morphaffixed onto the pronominal stem. We represent the basic type of pronouns in Hindi as π, which can be written as π_{np}, where $n = 1$ is singular, 2 is plural and $p = 1$ is first , 2 is second and 3 is third person.

Pronouns have a gender characteristic, but do not have a gender inflection. Therefore, while gender agreement rules exist for Hindi, it is not possible to ascertain whether a pronoun is masculine or feminine. Table 4.1 shows the various Hindi pronouns (no inflections) and their number and person.

	Singular	Plural
First	*mein* I/me	*hum* we/us
Second	*tum* you	*aap* you (pl. and hon.)
Third	*vaha* him/her/that	*ve* they/them/those

Table 4.1 Pronoun forms in Hindi

If affixed with a *vibhakti*, a simple type reduction takes place. An example pronoun and reduction is shown below.

Here the resulting type is the same as the pronoun itself, but the reduction is important for the genitive case construction. More importantly, a pronoun with a morphaffixed genitive marker behaves syntactically exactly like the marker itself, as it expects another noun or pronoun to its right [27, 104].

4.2.2 Nouns in Hindi

Nouns in Hindi behave much like pronouns, with the exception that they are not inflected upon. Nouns therefore show no inherent inflection to gender, though gender is an inherent property of nouns. Therefore nouns have gender, but no gender markers. This irregularity makes ascertaining agreement rules difficult. However, their gender is ascertained by using a lexicon, or using the gender marked by the genitive marker or the verb phrase. Therefore, to frame agreement rules for Hindi sentences, we propose providing gender subscripts to nouns if it can be directly ascertained from the lexicon. Noun phrases can be subjects, direct objects or other noun phrases such as predicate constructions. For the sake of generality, they have all been referred to as n here.

Nouns in Hindi have number, gender, and case characteristics. While a noun is always in third person, the transitivity of the verb and the *karaka* used affects the inflections on the common nouns. Therefore, they are represented by n_{ngc} where n is the number, $n = 1$ for singular and 2 for plural; g is the gender, $g = 1$ for masculine and 2 for feminine.

An example is presented here.

(10) gaaya soyii
cow.FEM. slept.FEM.PERF.
The cow sleeps.
$(n_{1\underline{2}13})(n^r_{1\underline{2}13}s_{12}) \rightarrow s_{12}$

karaka	*vibhakti*	**Equivalent Case**
karta	ϕ	Nominative
	ne	Ergative
karam	*ko*	Accusative
		Dative
karan	*se*	Instrumental
sampradan	*ko, ke liye*	Purpose/Reason
apadaan	*se*	Source
adhikaran	*me, par*	Locative

Table 4.2 Case/Role Marking in Hindi

In order to apply the above typing rules to proper nouns, note that proper nouns in Hindi do not have specific gender inflections, but still possess gender information. We shall take full examples of sentences in section 4.4.

4.2.3 Case Markers and *karaka*s

The Paninian system of *karakas* is akin to the the system of case in western lingusitics, as it shows the role of a noun to a particular verb in a sentence. Note that *karakas* are fundamentally relations that map semantic roles "demanded" by the verb to the morphological or lexical properties "provided" by the noun. The surface representation of these roles are done by markers known as *vibhaktis* which are detailed in Table 4.2.

The genitive case is not considered a *karaka* because it shows the link between two noun phrases. This relation is aptly named *sambandh* (literally *relationship*) [27]. Also unlike *vibhaktis*, the *sambandh* markers are inflected by the gender of the succeeding noun. As mentioned before, *sambandh* markers are gender marked, and therefore are represented as ρ_{ngc}, similar to the inflections for the noun mentioned above. The gender of the genitive marker is the gender of the noun that succeeds it.

For example:

(11) me -rii behan
I.SG. GEN. sister
My sister
$(\underbrace{\pi_{11}\ \pi^r}\ \underbrace{\rho_2})(\rho_2^r n_{12}) \rightarrow n_{12}$

We can generalize from the above subsections that the noun phrase in Hindi has gender, number, and person inflections and characteristics. The Hindi pronoun does not have a specified gender marker

(similar to the proper noun), while the noun itself is always in third person, and therefore does not have a dedicated person inflection or form. In order to generalize this notion, a basic type for the entire noun phrase can be introduced. This basic type, represented by N is subtyped by N_{gnpc}, where g, n, and p are gender, number, and person (which are represented using the same nomenclature as presented for π and n), while c is the subscript for noun form. $c = 1$ implies that the *vibhakti* is absent, while $c = 2$ implies the *vibhakti* is present in the noun phrase. This is only applicable for the nominal subject and direct object. For all other syntactic roles, c is not typed.

4.2.4 Adjectives

Adjectives are parts of speech which modify the noun by providing information about its shape, size, number, quality, quantity, or nature. Adjectives generally precede the noun they describe. In pregroup analysis, the basic type a is associated with adjectives. Adjectives in Hindi can be classified into two groups, inflected and uninflected.

4.2.4.1 Inflected Adjectives

As the name suggests, these adjectives are inflected for gender and number information. Therefore, these adjectives have the same subtyping rules as the nouns they precede, and must agree with them in gender and number, represented by a_{ngc} where the same convention for number and gender [74, 2].

For example:

(12) ba.rii la.rkii
big.FEM. girl
Big girl
$(a_{12})(a^r_{12}n_{121}) \rightarrow n_{12}$

The rules for adjective inflections for a noun in direct case are detailed below.

1. If the succeeding noun is masculine singular, the adjective is inflected with the morpheme *-aa*, for example: ***lambaa*** *la.rkaa* is a tall boy.

2. If the succeeding noun is masculine plural, the adjective is infected with the morpheme *-ey*, for example: ***lambe*** *la.rke* are tall boys.

3. If the succeeding noun is feminine, then irrespective of number, the adjective is inflected with the morpheme *-ii*, for example: ***lambii*** *la.rkii* is a tall girl.

In case of an oblique case, the corresponding adjective is inflected with *-e* for masculine singular as well.

4.2.4.2 Uninflected Adjectives

Uninflected adjectives do not represent any gender or number information. Such adjectives are therefore represented by a with no subscripts. In order to satisfy typing rules, an additional metarule is introduced, such that $a \rightarrow a_{ngc}$, i.e. an uninflected adjective can be reduced to **ANY** inflected adjectives, and can therefore reduce the adjoints of any inflection of this basic type.

For example:

(13) bahaadur la.rkii
brave girl
brave girl
$\underbrace{(a)}(a^r_{12}n_{12})$
$\rightarrow n_{12}$

4.3 Verb Phrase

In this section, the Hindi verb phrase has been introduced. First, some morphosyntactic properties of the verb are introduced, including the nature of tense and aspect marker, a single inflector function which controls the agreement inflections on the verb phrase, including the gender, number, person, and case indicators [94]. The inflector function is one of the most useful characteristics of pregroup calculus, as it combines the properties required for noun-verb agreement.

4.3.1 Properties of the Hindi Verb Phrase

The verb phrase in Hindi is a single unit that consists of a verb, an optional aspect marker and a tense marker. The verb phrase consists of either a single verb or a conjunct verb, complex verb or a light verb complex, which usually follows the construction "noun/adjective/verb + verbalizer" [3]. The tense markers and aspect markers are separate lexical items, while the future marker is a suffix on the trailing verb or verbalizer. The verbalizer interactions can be further classified based on their behavior with aspect markers such as infinitive + forms of *lagnaa* (to begin) [121, 44].

Note that following the rules of morphological type reduction, the morphaffixed aspect marker are also given individual basic types. The main verb in a sentence is given a basic type s. We introduce a type ν for verbs which are not the main verb. s and ν typed verbs behave similarly, except that the ν type verbs do not require tense makers, as their tense is determined by the tense of the sentence (main verb). ν can also be used to represent verbs in light verb constructions.

The verb itself is unmarked, but the aspect marker is inflected by gender, number and person. [97, 120] explains the rules of gender, number and person marking on tense and aspect markers in detail, which may be summarized as follows:

- The aspect marker inflections are summarized as:

1. If the noun inflecting the verb phrase is in second person, the plural form is not gender marked and the singular masculine form is the same as the plural form.
2. The form of the aspect marker is the same in both first and third person for all the forms, and it is inflected by gender as well as number.
3. The form of the aspect marker is the same if the noun inflecting the verb is singular and feminine.
4. The perfective aspect requires the nominal subject in ergative case, while the habitual and progressive require the subject in a nominative case.

- The tense marker inflection are summarized as:

1. The present tense marker is inflected by the person and number, but not by gender of the noun that inflects the verb phrase.
   - The present tense marker has the same form for singular and plural if the noun that is inflecting it is in the second person.
   - The present tense marker has the same form for first and third person if the noun inflecting it is plural.
2. The past tense marker is inflected by the gender, number and person of the noun that inflects the verb phrase.
3. The future tense marker is inflected by the gender, number and person of the noun that inflects the verb phrase.

We represent tense suffixes and markers as τ_t where $t = 1$ is past tense, 2 is present, 3 is future. Similarly, the aspect marker α_i is now subtyped with $i = 1$ for the perfect form, 2 for the imperfect form and 3 for the continuous form.

Due to complications in representing an agreement structure that accounts for the irregularity of constraints as mentioned above, the concept of the inflector function has been slightly modified for verb representation in Hindi. Instead of being the representation of an abstraction over the infinitive form of the verb, the function can be seen as a typed function from the basic types to the subtype of the basic types, and the type of the function can be used for agreement with the noun phrase. Therefore the inflector function is represented by C_{ngp} where n stands for number, g for gender and p for person. The system of numbering is the same as that of the noun phrase; i.e. $n = 1$ is singular and $n = 2$ is plural; $g = 1$ is masculine and 2 is the feminine form; and $p = 1$ is first person, 2 is second and 3 is third person.

There is a possibility that the verb form does not contain any particular gender, number or person information, such as the present tense marker *hai* (is). In this case, we shall use the symbols n, g or p to denote these general forms. For example:

(14) vaha ghar jaa -tii thii
They.SING home go PERF.FEM was.FEM

She was going home.

$(\underbrace{\pi_{13}}) \ (\underbrace{o_{11}}\underbrace{(o^r}\pi^r_{1p}s\tau^l)\underbrace{(\alpha_1)}\underbrace{(\alpha^r}\tau_1) \rightarrow s$

There are three important observations to be made here. First, the tense marker is agnostic to the form of the aspect marker, as the tense marker does not change based on the aspect marker form. Secondly, morphological type reduction is used here by treating the morph-affixed aspect marker. Lastly, the object adjoint o^r is not inflected by an inflector function.

While the inflector function C_{ngp} represents the characteristics of the verb phrase, the noun phrase also has the same characteristics which can be abstracted out. In the subsequent subsections, we discuss the formalization and representation of two common verb constructions in Hindi, causative and light verb constructions.

4.3.2 Light Verb Constructions

A light verb is defined as a complex predicate construction where a noun or verb is in joint predication with a verbalizer which takes the tense and the aspect markers, making the joint predication a semantic whole [26, 125]. Light verbs in Hindi are non-compositional, where the meaning of the joint predicate is derived from the meaning of participating noun or verb, while the verbalizer, while syntactically similar to a main verb, acts as a syntactic bridge.

From a formalisms perspective, the representation of light verbs is a challenge. On one hand, the verbalizer can be represented as the main verb and given the type s, which is syntactically accurate, but would imply treating the noun or verb phrase as an argument of the verb, which is semantically incorrect. On the other hand, since the combination of the noun or verb phrase and the verbalizer is the main verb, the entire phrase can be given the type s, which is syntactically inaccurate as the noun can contain gender information which affects the gender inflection of the tense and aspect markers.

Therefore, in order to represent light verb constructions in an syntactically accurate and a semantically coherent manner, we introduce a joint predication of a noun phrase n or a verb phrase ν with a verbalizer, also represented by ν such that:

$$\underbrace{(n \cdot \nu)}_{s} \rightarrow s \qquad \underbrace{(\nu \cdot \nu)}_{s} \rightarrow s$$

The inflector function is applied to the entire joint predicate, so the gender of the participating noun gets "overwritten" by the gender of the inflector (gender of the tense and aspect marker).

(15) us -ne mujh -e yaad ki -yaa
They.SING. NOM. me ACC. remember do PERF.MASC.
He/she remembered me.

$(\pi_{12}\underbrace{\kappa^l_1})\underbrace{(\kappa_1)}(\pi_{11}\underbrace{\kappa^l_2})\underbrace{(\kappa_2)}(\pi^r\pi^r\underbrace{n \cdot \nu}_{s}\underbrace{\alpha^l_2})\underbrace{(\alpha_2)}$

$\rightarrow (\underbrace{\pi_{12}})\underbrace{(\pi_{11})}\underbrace{(\pi^r}\pi^r s)$

$\rightarrow s$

Note that in the above example, the verb is inflected with the default gender, number and person so the entire verb phrase remain uninflected thus, no inflector function has been used. The main verb s remains untyped.

4.3.3 Causative Verbs

A causative construction is defined by the transfer of agency of the action from the subject to the direct object, where the degrees of transfer determine the degree of the causative constructions. Early literature on Hindi verb constructions consider the concept of a surface agent, recipient and mediator [67, 73]. From a lexical formalism perspective, causatives in Hindi are known by their suffixes; *-A* is used for first degree causative and *-vA* for second degree. Note here that the causative construction can be used with tense and aspect markers just like any other construction.

In order to represent a causative construction, we introduce a new basic type c_i where $i = 1$ refers to first degree of causative and 2 refers to a second degree causative. For example:

(16) vaha mujh -se kaam kar -vaa -tii thii
They.SING. me DAT. work do CAU. PERF. FEM. was.FEM.
She used to make me do work.
$(\pi_{13})(o_{11})C_{12p}\{(o^r\pi^r s\ \underbrace{\tau^l})(\underbrace{\alpha_1})(\underbrace{\alpha^r\tau_1})\}$

$\rightarrow (\underbrace{\pi_{13}})(\underbrace{o_1})(\underbrace{o^r\pi^r_{1p}}s_{12p})$

$\rightarrow s_{12p}$

We see here that the causative construction is reduced by using order alternation of the pregroup elements [42, 54]. This is because the causative construction is a verb characteristic, and can not be reduced by the tense and aspect markers. It is also used to represent the agent of the action is not the same as the agent transferring the action. Therefore, the agent transferring the action will reduce the causative construction, along similar lines as done by [30].

A summary of the subtypes of all the Basic Types: s_{ngp} being sentence, n_{ngc} a subject noun phrase, o_{ng} a direct object, a_{ng} an inflected adjective (and a an uninflected adjective), p_{ng} a predicate, π_{np} a pronoun, α_a an aspect marker, τ_t a tense marker, c_i a causative construction, C_{ngp} an inflector function, and N_{ngp} the nominal inflector function. The order in which they have been written is the number used to denote that property.

4.4 Examples of Sentences

In this section, we take examples of basic Hindi sentences to show the reduction of grammatical sentences based on the constraints and their representation provided above. The sentences have been

Property		Gender g	Number n	Person p	Tense t	Aspect a	Degree i	Case c
For		n, o, p, ρ π, C, s, a	n, o, p, ρ π, C, s, a	π, C, s a, ρ	τ	α	c	n, a, ρ
Subtypes	1 = 2 = 3 =	masculine feminine	singular plural	first second third	past present future	perfect habitual continuous	first second	direct oblique

Table 4.3 Subtypes for all basic types

adapted from the Hindi Dependency Treebank [18]. A summary of the basic types, their subtypes and the inflector function introduced in the paper are provided in Table 4.3. Through this section, we follow alternation of basic types based on rules established in [42] and [54].

Given the set of basic types $\{\pi, s, p, a, c, o, n, \kappa_i, \rho, \alpha, \tau\}$, simple sentences can be typed in Hindi as follows:

(17) mei.m slool jaataa hu.m
I.NOM skool.ACC go.HAB is
I go to school.
$(\pi_{g110})(o_{g110})C_{1110}\{(o^r\pi^r s\tau^l)(\alpha^l\ \alpha_2\ \tau_1)\} \rightarrow s$

(18) .tiinaa ne gaanaa gaayaa
Tina.ERG. CASE song.ACC. sang.PFV
Tina sang a song.
$(n_{2131}\ \kappa_1{}^l)(\kappa_1)(o_{1130})C_{1131}\{(o^r n^r s\alpha^l\alpha_1)\} \rightarrow s$

(19) raam ne gend ko balle se maaraa thaa
Ram.ERG ERG. MARKER ball.ACC ACC. MARKER bat.INST. with hit.PFV. was
Ram hit the ball with a bat
$(n_{1131}\kappa_1{}^l)(\kappa_1)(o_{1131}\kappa_2{}^l)(\kappa_2)(p_{113c}\kappa_3{}^l)(\kappa_3)C_{1131}\{(p^r o^r n^r s\ \tau^l\alpha^l\alpha_1)(\tau)\} \rightarrow s$

4.5 Conclusion

In this chapter, the Hindi noun and verb phrase are introduced. I detail some of the rules for typing the noun phrase, the verb phrase, and its various constituents. In formalizing the grammar by constituents, there are also a large number of global characteristics (such are gender, number, person, and case) which

govern the agreement rules between the noun and the verb phrase in Hindi, as well as characterize the use of markers, such as postpositional *vibhaktis* in the subject and direct object role.

The formalization presented here is in no way the entire Hindi grammar. The Hindi verb phrase is far more complex, and includes various moods and modal markers in verb phrases, and honorific and formal forms and rules for nouns and verbs both. However, the preliminaries of the Hindi grammar detailed and typed in this chapter are for an introductory formalism of the syntactic structure using pregroups, as has been done by existing literature for various languages detailed in Chapter 2.

Finally, the nature of inflectional markers in Hindi poses a challenge to the aim of pregroup calculus to provide a strictly lexicalized grammar of the language. *vibhaktis*, *sambandh*, tense, and aspect markers can either be independent morphemes (for example, the continuous marker *rahaa*), dependent morphemes (for example, the habitual marker *taa*), or either depending on the part-of-speech of the word preceding it (for example, *vibhaktis* with nouns are independent, while with pronouns are dependent morphemes). While this chapter dealt with such instances as "morphosyntax", the next chapter identifies and rectifies the formalism in order to make it a uniform representation.

The sentences selected above are those which are easily be represented with the type signatures provided above. However, these examples are not as complex as sentences which occur in "real world" scenarios, sentences with embedded clauses, subjunctives, and basic types beyond the ones that are not mentioned in the sentences we have seen thus far. The expansion of the basic type list, as well as the analysis of complex syntactic constructions remains a topic for future work.

Chapter 5

Representing Syntactic Properties

The previous chapter introduced the preliminary formalism of Hindi noun and verb phrases and their various constituents. However, syntactic structure is more than simply the relationship between the constituents and their interaction. In this chapter, I discuss some of the most important characteristics of Hindi sentence structure at two levels.

First, I introduce morphological type reduction, a mechanism to uniformly represent clitics *within* the noun and verb phrase in Hindi. This view of pregroup calculus allows for a morphosyntactic grammar which identifies both inflections and lexicalized function words as equal contributors in terms of basic type assignment and reduction, as well as provides uniformity to the grammar.

Second, I elaborate upon the nature of word order in Hindi. Hindi has a relatively free word order, wherein constituents can be shuffled due to a combination of explicit role markers and the aspect marker enforcing case constraints on the noun phrases. However, shuffling of words inside a constituent is not allowed. While other categorial formalisms do not explicitly handle word order shuffling, pregroups only developed a mechanism of complete free word order movement. I extend that notion using the principles presented in Chapter 3 to introduce partial word order shuffling, and methods of enforcing that for such languages. These methods can be used for other languages as well, including adverbs and prepositional phrases in English.

Finally, I introduce the notion of subtyping and agreement rules for robust type checking and formal parsing of Hindi syntax. Based on the rules defined before, the grammar introduces some language-specific metarules for reduction. These metarules are designed to allow reductions in cases wh ere possible basic types could suffice, for example, uninflected and inflected adjectives, or a pronoun in place of a noun phrase.

5.1 Morphological Type Reduction

Pregroup representations are lexical, therefore, each word is provided a compound type. However, as seen above, the case markers as well as some aspect markers can be morphologically affixed [121, 5]. These marker are not clitics, as they are not phonetically dependent on the surrounding morphemes,

but can be morphologically bound to the previous word in some cases, and be entirely free in others. Therefore, in order to deal with a uniform pregroup representation for these markers, we introduce the concept of morphological type reduction. The basic type provided to a word with a morphaffixed case marker or verbal auxiliary will be derived from the concatenation and reduction of the basic types provided to the stem and the marker individually. This is seen in first person and second person pronouns as well as habitual and perfective aspect and future tense constructions. Examples in chapter 4 are already morphologically type reduced, but we present explicit examples here.

In short, *morphological type reduction* is a justification for the type given to a word in a sentence, which is composed of morphemes which could be individually typed, if they occurred as words in a sentence. This has been done for case marking in Sanskrit [42]. In the following paragraphs, the morphological type reductions of personal pronouns and personal possessive pronouns are provided. Note that morphological type reduction is done only for the inflectional which exists as a lexical marker in the Hindi lexicon, or if other units of that basic type are lexical items.

Perfective verbs and habitual verbs occur with the suffixes *-aa* and *-taa* respectively. For example, the verb:

(20) maaraa
hit.PERF.
to hit
$(s\ \underbrace{\alpha^l})\ (\alpha_1) \rightarrow s$

The verb *maaraa* is treated as *maar* + *aa*, wherein the aspect marker *A* is given the type α_1, the perfective aspect. While the perfective marker is morphaffixed, the continuous aspect marker is a lexical marker, not a bound morpheme. Therefore for the same verb:

(21) maar rahaa
hit CONT.
hitting
$(s\ \underbrace{\alpha^l})\ (\alpha_3) \rightarrow s$

Personal pronouns to which the *karaka* marker is affixed are given the type (π_{gnp}). For example, the morphological type reduction of the word:

(22) maine
I.ERG.
I (as a subject)
$(\pi_{g11}\ \kappa^l)\ (\kappa_1) \rightarrow \pi_{g11}$

would be typed as *mein* + *ne*. Personal possessive pronouns, however, are typed the same as the genitive case marker ρ, rather than the type of the personal pronoun π. The morphological type reduction of the word:

(23) meraa
I.GEN.MASC.

my

$(\pi_{g11})(\pi^r\ \rho_1) \rightarrow \rho_1$

Note here that the resulting type of a pronoun with an *sambandh* is the type of the *sambandh* marker itself. This is because the gender inflection of the marker depends on the noun to which it is associated.

In summary, in order to develop a system of uniform and consistent type assignment and analysis, the pregroup grammar of Hindi is not entirely lexical in nature. While in chapter 4 light verbs were treated as a single syntactic constituent despite being represented using more than one word, in this section, all bound morphemes which have a free form in the same syntactic category, or are free if associated with a different part-of-speech, are treated the same. Therefore, the type grammar of Hindi can be considered both morpholexical rather than strictly lexical.

Other languages have aimed to deal with clitics, including Italian and Persian [40], by differentiating the citicized and other basic types. Clitics in these languages are also the basic types involved in word order alternation. However, as discussed before, Hindi is a relatively free word order language. This implies that freedom of movement is not restricted to just a specified set of basic types. More specifically, the change in word order is rooted in constituents, not words. Due to this, creating an alternate set of basic types for every possible transformation of a sentence tedious and inefficient. However, both Italian and Persian, as well as Hungarian (which also has a predominance of clitics), can have a morphosyntactic type system which creates a consistent typing for inflections to deal with uniform typing under word order alternation.

In the section below, the representation of Hindi word order is introduced. Using the properties of precyclic pregroups, I showcase the idea of free movement of words. I then identify the possible errors in free word order alternation, namely that it parses grammatically incorrect sentences due to restrictions in shuffling word order in Hindi grammar. Finally, I provide three methods of dealing with it.

5.2 Representing Word Order

In order to apply precyclicity rules, the example chosen is a simple transitive verb from the sentence "Tina sang a song", which has been typed above in chapter 4. The procedure for applying the rules has been as described by [42] for Sanskrit. As above, the arcs denote a reduction, while the underline shows the arguments of the precyclic transformation. In the exmaples below, subtype analysis is ignored for brevity.

(24) gaanaa .tinaa ne **gaayaa**
song Tina.ERG CASE sang
Tina sang a song.

$(o)(n\kappa_1{}^l)(\kappa_1)(o^r n^r s)$
$\rightarrow (o)\overline{(n)(\underline{o^r n^r} s)}$
$\rightsquigarrow^{ll} (o)(n)(n^r o^l s)$
$\rightarrow \underline{(o)(o^l s)}$
$\rightsquigarrow^{rr} (o^l s)^{rr}(o)$
$\rightarrow (\underline{o^r s^{rr}})(o)$
$\rightsquigarrow^{ll} (s^{rr} o^l)(o)$
$\rightarrow s^{rr}$
$\rightsquigarrow^{ll} s$

An example of a sentence with two movements is as follows, for the sentence "Ram had hit the ball with a bat".

(25) balle se gend ko raam ne maaraa thaa
bat.INST. CASE ball.ACC. CASE ram.ERG. CASE hit.PERF. did
Ram hit the ball with a bat

$(p\kappa_3{}^l)(\kappa_3)(o\kappa_2{}^l)(\kappa_2)(n\kappa_1{}^l)(\kappa_1)(p^r o^r n^r s\ \tau^l)(\tau)$
$\rightarrow (p)(o)(n)(p^r \underline{o^r n^r} s)$
$\rightsquigarrow^{ll} (p)(o)(n)(\underline{p^r n^r} o^l s)$
$\rightsquigarrow^{ll} (p)(o)(n)(n^r p^l o^l s)$
$\rightarrow (p)(o)(\underline{p^l o^l} s)$
$\rightsquigarrow^{rr} (p)(o)(o^r p^l s)$
$\rightarrow \underline{(p)(p^l s)}$
$\rightsquigarrow^{rr} (p^l s)^{rr}(p)$
$\rightarrow (\underline{p^r s^{rr}})(p)$
$\rightsquigarrow^{ll} (s^{rr} p^l)(p)$
$\rightarrow s^{rr}$
$\rightsquigarrow^{ll} s$

Note that in the first example, the movement was *gaanaa* and *tinaa* ***ne***, and not just *tinaa*; similarly, in the second sentence, the constituents being shuffled are *balle se* and *gend ko*. Hindi allows only constituent scrambling, the pregroup grammar has to account for this restriction. For example, the following construction should **NOT** be allowed:

(26) .tinaa gaanaa ne gaayaa
Tina.ERG. song.ACC. CASE sang
Tina sang a song
$(n\kappa_1^l)\underline{(o)(\kappa_1)}(o^r n^r s)$
$\rightsquigarrow^{ll} (n\underline{\kappa_1{}^l)(\kappa_1})(o^{ll})(o^r n^r s)$
$\rightarrow \underline{(n)(o^{ll})}(o^r n^r s)$
$\rightsquigarrow^{rr} (o)(n)(\underline{o^r n^r} s)$
$\rightsquigarrow^{ll} (o)(\underline{n)(n^r} o^l s)$
$\rightarrow \underline{(o)(o^l s)}$
$\rightsquigarrow^{rr} (o^l s)^{rr}(o)$
$\rightarrow (\underline{o^r s^{rr}})(o)$
$\rightsquigarrow^{ll} (s^{rr}\underline{o^l)(o})$
$\rightarrow s^{rr}$
$\rightsquigarrow^{ll} s$

As seen above, the current pregroup grammar rules allow for the reduction of sentences which are disallowed by the grammar. This section explains the methods taken to restrict word movement, in order to allow only constituent scrambling, keeping the order of the words within a constituent constant.

5.2.1 Pregroup Grammar Rules

Specific changes in word order are disallowed by Hindi grammar, such as inserting a noun within a noun constituent, as was done in the example *tinaa gaanaa ne gaayaa*, or inserting a noun or noun phrase in the verb constituent. Therefore, the pregroup grammar has to restrict certain transformations, such that such reductions are not possible. The pregroup grammar has to encode such restrictions in word order alternation. For example, if the rule reads "The alternation of ANY phrase with a *karaka* marker is disallowed", it may be represented in the following way:

$$\forall x \in \mathcal{B}$$
$$(x)(\kappa_i) \not\rightsquigarrow^{ll} (\kappa_i)(x^{ll})$$
$$(x)(\kappa_i) \not\rightsquigarrow^{rr} (\kappa_i^{rr})(x)$$

where, as discussed in Chapter 3, $\mathcal{B}$ is the set of all basic types in the language. A similar set of rules can be created for alternation in verb constituents, where no word order alternation is allowed between a verb and its aspect marker, and between the aspect and the tense marker, mirroring the rules of the language.

$$(s)(\tau) \not\rightarrow^{ll} (\tau)(s^{ll}) \quad (s)(\tau) \not\rightarrow^{rr} (\tau^{rr})(s)$$
$$(\tau)(\alpha) \not\rightarrow^{ll} (\alpha)(\tau^{ll}) \quad (\tau)(\alpha) \not\rightarrow^{rr} (\alpha^{rr})(\tau)$$
$$(s)(\alpha) \not\rightarrow^{ll} (\alpha)(s^{ll}) \quad (s)(\alpha) \not\rightarrow^{rr} (\alpha^{rr})(s)$$

Therefore, in the example *tinaa gaanaa ne gaayaa* (Tina song erg. sang) presented above, we have:

$$(n\kappa_1^l)\underline{(o)(\kappa_1)}(o^r n^r s)$$
$$\not\rightarrow^{ll} (n\underline{\kappa_1{}^l)(\kappa_1)}(o^{ll})(o^r n^r s)$$

Thus the sentence will not reduce to s as it is deemed ungrammatical.

5.2.2 Selective Transformation

Selective transformation is a procedure that disallows the precyclic transformation of a step in the reduction, provided that some elements of the pregroup representation belong to a set that does not allow precyclic transformation, hence allowing only a selective application of the precyclic conversion rules.

As discussed in section 3, $\mathcal{B}$ is the set of all basic types in the language, and $T(\mathcal{B})$ is the free pregroup over that set. In order to apply selective transformation, two sets The sets $\mathcal{B}_T$ and $\mathcal{B}_{NT}$ are defined, such that the set of all basic types $\mathcal{B} = \mathcal{B}_T \cup \mathcal{B}_{NT}$, where $T(\mathcal{B}_{NT})$ is a free, proper, non-precyclic pregroup, while $T(\mathcal{B}_T)$ is a proper, free, precyclic pregroup. The union of a precyclic and a non-precyclic pregroup is a non-precyclic pregroup, and no other properties of the pregroup are affected.

Within the examples seen above, the basic types of the *karaka* marker and *sambandh* marker in the noun phrase, and the tense and aspect marker in the verb phrase should not ll- or rr-transformed, while the personal pronoun, sentence, predicate, noun phrase, and direct object types are allowed to transform. Therefore, $\{\kappa_i, \rho, \alpha, \tau\} \in \mathcal{B}_{NT}$ and $\{\pi, s, p, o, n\} \in \mathcal{B}_T$. Therefore, to apply transformation rules, only the types in the sentence should be those belonging in $\mathcal{B}_T$. Therefore, in the example *tinaa ne gaanaa gaayaa* (Tina erg. song sang), we have:

$$(o)(n\underline{\kappa_1{}^l)(\kappa_1}\,)(o^r n^r s)$$
$$\rightarrow (o)(n)(\underline{o^r n^r} s)$$

where all the elements in the second line belong to $\mathcal{B}_T$, which allows the transformations and reductions:

$$\rightsquigarrow^{ll} (o)(\underline{n)(n^r} o^l s)$$
$$\rightarrow \underline{(o)(o^l s)}$$
$$\rightsquigarrow^{rr} (o^l s)^{rr}(o)$$
$$\rightarrow (\underline{o^r s^{rr}})(o)$$
$$\rightsquigarrow^{ll} (s^{rr} o^l)(o)$$

Constituent Type	Constituent Profile
Subject	$(x\kappa_1^l)(\kappa_1) \rightarrow x$ for $x \in \{\pi, n\}$
Direct Object	$(o\kappa_2^l)(\kappa_2) \rightarrow o$
Predicate	$(p\kappa_i^l)(\kappa_i) \rightarrow p$ for $i \in [3, 6]$
Nominal Relations	$(\rho)(\rho^r x) \rightarrow x$ for $x \in \{n, o, p\}$ $(n)(n^r\rho)(\rho^r x) \rightarrow x$ for $x \in \{n, o, p\}$
Verb Forms	$([x^r]^+ s\alpha^l)(\alpha\tau^l)(\tau) \rightarrow ([x^r]^+ s)$ for $[x] \in \{n, o, p\}$ $([x^r]^+ s\tau^l)(\tau) \rightarrow ([x^r]^+ s)$ for $[x] \in \{n, o, p\}$
Sentence	$(n)([x])^+([x^r]^+ n^r s) \rightarrow s$ for $[x] \in \{o, p\}$

Table 5.1 Constituent Profiles for Hindi

$$\rightarrow s^{rr}$$
$$\rightsquigarrow^{ll} s$$

while in the example of the construction *tinaa gaanaa ne gaayaa* (Tina song erg. sang) presented above, we have:

$$(n\kappa_1^l)\underline{(o)(\kappa_1)}(o^r n^r s)$$

which is not reducible as the transformations cannot be applied. Therefore ungrammatical reductions are disallowed.

5.2.3 Two-Step Reduction

As mentioned above, Hindi allows constituent scrambling as opposed to word order scrambling. Therefore the reduction of a sentence can be deconstructed into two steps, the reduction of constituents, followed by reduction of the sentence. This process guarantees that ungrammatical reductions will not take place.

For the two-step reductions, first the "constituent profile" has to be defined. A constituent profile is a general construction to which constituents in Hindi can be mapped. Each constituent profile has a reduction which can be applied to a sentence. A constituent profile is specific to the type of phrase expected. A sequence of words which does not follow any constituent profile cannot be reduced. This will disallow the reduction of ungrammatical sentences.

Table 5.1 shows the constituent profiles for the examples provided above. The $[x]^+$ represents one or more elements of the basic type x. Note that there is no need for a transformation in any of the constituent profiles, as it reduces to the constituent $\underline{\text{head}}$ form automatically. The constituents can be

nested, and the constituent profile reflects this. A sentence may be defined as a specific case of a constituent profile, as it is the *only* profile which allows transformations before reductions.

Two-step reduction can be achieved as follows:

- **Type the sentence**: This allows the recognition of all possible constituent profiles, as well as conflicts with the constituent profiles.
- **Isolate and reduce constituents**: The constituents are mapped to the constituent profiles, and reduced accordingly.
- **Replace reduced forms in the sentence**: The constituents, once reduced, are placed back into the order in which they occurred in the original sentence. The sentence form should resemble the "sentence" profile.
- **Transform and reduce**: The sentence is then transformed and reduced according to the rules of transformation, as has been done above.

An example of two-step reduction for the sentence *raam ne gend ko balle se maara thaa* (Ram hit the ball with a bat) would be as follows:

Step 1:

$$\underline{(n\kappa_1^l)(\kappa_1)}\ \underline{(o\kappa_2^l)(\kappa_2)}\ \underline{(p\kappa_3^l)(\kappa_3)}(p^r o^r n^r s\underline{\tau^l)(\tau)}$$

Step 2: $(n\kappa_1^l)(\kappa_1) \rightarrow n$, according to the subject constituent profile. Similarly, $(o\kappa_2^l)(\kappa_2) \rightarrow o$ and $(p\kappa_3^l)(\kappa_3) \rightarrow p$ are also valid reductions according to the object and predicate profiles. There is also the $([x^r]^+ n^r s\tau^l)(\tau) \rightarrow ([x^r]^+ n^r s)$, which is a valid verb form reduction.

Step 3: $(n)(o)(p)(p^r o^r n^r s)$ is obtained, which fits the sentence profile.

Step 4: $(n)(o)(p)(p^r o^r n^r s) \rightarrow s$, which is the required reduction

An incorrect reduction can be recognized easily. Given the example that was provided in Section 4 *tinA gAnA ne gAyA*. After step 1, the following is obtained:

$$\underline{(n\kappa_1^l)(o)(\kappa_1)}(o^r n^r s)$$

which cannot be found in any constituent profile. Therefore, the sentence is ungrammatical, and will appropriately not be represented by the grammar.

5.2.4 Beyond Hindi Syntax: Word Order and English

Lambek's work in English type grammar [82] and the work that has followed [109, 122] have not dealt with word order alternation in English yet. While English is a relatively fixed word order language, note that prepositional phrases and adverbial phrases are relatively free, especially with intransitive verbs. Therefore, while the default order remains "Subject-Adverb-Verb-PP", it can be seen in the following sentences that this word order can also change.

- Default (S-Adv-V-PP): "I quickly ran into the fields."
- PP-S-Adv-V: "Into the fields, I ran quickly."
- Adv-S-V-PP: "Quickly I ran into the fields."
- S-V-Adv-PP: "I ran quickly into the fields."
- S-V-PP-Adv: "I ran into the fields quickly."

To develop a robust representation of English word order, first, the initial word order constraints must be noted. The standard word order in English is SVO, which reduces to SV in the case of intransitive verbs, which is the focus of this sample. We examine the word order alternation in this fragment of English, using the methods explored in the paper.

First, the set of basic types $\{\pi, o, s, n, p\}$ has to be expanded to include types for adverbial phrase A and type for prepositional phrase ρ. The subscripts which characterize gender, number, person or tense have been ignored for this example. The determiner is considered a part of the noun phrase. In the exmaples provided below, subtypes are not considered. Therefore, the sample sentence, in the default word order, can be typed as follows ("the fields" has been typed p):

1.

I	quickly	ran	into	the fields.
π	A	$A^r\pi^r s\rho^l$	ρp^l	p

$$(\underbrace{\pi})(\underbrace{A})(\underbrace{A^r\pi^r s\rho^l})(\underbrace{\rho p^l})(\underbrace{p}) \to s$$

This reduction is reasonably straightforward. On changing the word order, precyclic transformations are applicable, so a similar reduction for the statement, "Quickly I ran into the fields", can be handled as follows:

2.

Quickly	I	ran	into	the fields.
A	π	$A^r\pi^r s\rho^l$	ρp^l	p

$$\begin{gathered}(A)(\pi)(A^r\pi^r s\underbrace{\rho^l})(\rho\,\underbrace{p^l})(p)\\ \to (A)(\pi)(\underline{A^r\pi^r}s)\\ \rightsquigarrow^{ll} (A)(\underbrace{\pi})(\underbrace{\pi^r}A^l s)\\ \to \underline{(A)(A^l s)}\\ \rightsquigarrow^{rr} (\underline{A^r s^{rr}})(A)\\ \rightsquigarrow^{ll} (s^{rr}\underbrace{A^l})(\underbrace{A}) \rightsquigarrow^{ll} s\end{gathered}$$

The sentence "Quickly I ran the fields into" is an ungrammatical sentence and should not be reducible by the grammar. However:

3. *

Quickly	I	ran	the fields	into.
A	π	$A^r\pi^r s\rho^l$	p	ρp^l

$$
\begin{aligned}
&(A)(\pi)(A^r\pi^r s\rho^l)(p)(\underline{\rho p^l}) \\
\rightsquigarrow^{rr}\ &(A)(\pi)(A^r\pi^r s\underline{\rho^l})(\underline{p})(\underline{p^r}\rho) \\
\rightarrow\ &(A)(\pi)(\underline{A^r\pi^r s}) \\
\rightsquigarrow^{ll}\ &(A)(\underline{\pi})(\underline{\pi^r} A^l s) \\
\rightarrow\ &\underline{(A)(A^l s)} \\
\rightsquigarrow^{rr}\ &(\underline{A^r s^{rr}})(A) \\
\rightsquigarrow^{ll}\ &(s^{rr}\underline{A^l})(\underline{A}\ ) \rightsquigarrow^{ll} s
\end{aligned}
$$

Therefore, the methods discussed in sections above can be used to disallow this reduction.

- Using the method of restriction of pregroup grammar rules, the transformation $(\rho p^l) \rightsquigarrow^{rr} (p^r\rho)$ is disallowed. So the order of the prepositional phrase and its predicate remains the same, disallowing an ungrammatical reduction.
- Two-step reduction can be used to establish the prepositional phrase profile as $(\rho p^l)(p)$, and the sentence does not follow this profile. Therefore, further reduction is disallowed.
- Under selective transformation, the only basic type that can belong to $\mathcal{B}_{NT}$ is p. Therefore, its alternation with the type ρ is considered ungrammatical and the sentence is not reduced.

Hence, all three methods can be used to disallow the representation and reduction of ungrammatical sentences by applying the relevant word order constraints to the freedom in word order.

5.3 Agreement Rules: An Analysis into Subtyping

So far, two major facets of a robust and uniform syntactic representation have been discussed which are not only relevant for Hindi grammar analysis, but are also applicable to other languages and how clitics are handled in the pregroup framework. However, all the examples so far have been well typed due some underlying principles about the relationship between various constituents in the language. In this section, I make these assumptions explicit by focusing on subtyping and its role as an indicator of feature agreement. I also explore some language specific metarules that allow rules for the relationship between subject and verb to become more explicit and therefore parsable.

Chapter 4 provides a detailed list of rules for the inflection of the tense and aspect marker in Hindi based on the gender, number, and person characteristics of the relevant noun phrase. Using a single inflector function for the verb phrase allows a single function to capture this information for the entire

verb phrase. In the same vein, the nominal inflector N has been introduced as well, which is useful to abstract out the characteristics of the noun phrase as well, which include whether or not a case marker is used.

The agreement of gender, number, and person between the noun and the verb phrase can be demonstrated using the three examples below:

(27) ladkiyon ne paanii piyaa
girls.FEM.PL.ERG. CASE water.SG.ACC. drink.PERF.MASC.
The girls have drunk water.
$(n_{2231}\underline{\kappa_1^l})(\underline{\kappa_1})(\underline{\ o_{1130}})\ \underline{C_{1131}\{o^r\ n^r}\ s\underline{\alpha^r}\ \underline{\alpha_1}\} \rightarrow s_{1131}$

(28) ladkiyaan paanii piitii hain
girls.FEM.PL.NOM. water.SG.ACC. drink.PERF.FEM. be.PRES.PL.
Girls drink water.
$\underline{(n_{2230})(o_{1130})C_{2230}\{(o^r\ n^r}\ s\underline{\tau^l}\ \underline{\alpha^l\ \alpha_1})(\underline{\tau_1})\}$

In the examples above, it is visible that the case of the subject (nominative or ergative) determines whether the verb phrase will have default inflections, or those which agree with the nominal subject. Furthermore, different aspects enforce conditions on the use of *vibhaktis* for the nominal subject. Therefore, agreement of the verb phrase is generally with that noun phrase which does not use an *vibhakti*. In case neither subject nor direct object use it, the subject takes precedence, while if both use a *vibhakti*, then the default male singular third person inflection is used for the entire verb phrase.

While the use of the *ne vibhakti* for the nominal subjects is based entirely on the aspect marker of the main verb, the use of the *ko vibhakti* is more complex, as it depends on factors such as the animacy of the object. Hence, it is beyond the scope of this thesis. However, the above rule still applies, i.e. in the case that the sentence is ergative and accusative (no case marker on the direct object), then the verb agrees with the inflections of the direct object. For example:

(29) raam ne saaikil chalaayii
Ram.ERG. CASE cycle.SG.FEM.ACC. ride.PERF.SG.FEM.
Ram rode a cycle.
$\underline{(n_{1131}\underline{\kappa^l})(\underline{\kappa_1})(o_{2130})C_{2131}\{(o^r}\ \underline{n^r}\ s\underline{\alpha^l})(\underline{\alpha_1})\}$

(30) raam ne saaikil ko chalaayaa
Ram.ERG. CASE cycle.SG.FEM.ACC. CASE ride.PERF.SG.MASC.
Ram rode a cycle.
$\underline{(n_{1131}\underline{\kappa^l})(\underline{\kappa_1})(o_{1131}\underline{\kappa^l})(\underline{\kappa_2})\ C_{1131}\{(o^r}\ \underline{n^r}\ s\underline{\alpha^l})(\underline{\alpha_1})\}$

While the sentences above are semantically equivalent, the inflections on aspect marker differ based on the presence or absence of the *ko vibhakti* (accusative marker). In the type analyses provided above, the characteristic that stands out is that in the presence of a case marker, the N is presented with a ${}^{b}ot$ symbol, for the subject and the direct object. For ditransitive verbs, the indirect object is always succeeded by a *vibhakti*, while all other predicates and their respective *karakas* are generally succeeded by their *vibhaktis* as well. Therefore, the distinction is generally provided only to the subject and direct object.

While gender agreement also applies to the adjective and the succeeding noun, uninflected adjectives do not change based on the features of the noun, while inflected adjectives do. Introducing different basic types for the same syntactic notion is overbearing on the type system, and therefore we introduce the following is metarule:

$$a \to a_{ngc}$$

According to the metarule above, an uninflected adjective can be reduced to an inflected adjective for any and all number, gender, and case modifiers. This implies that if regardless of the nature of the noun, an inflected adjective can modify it. Also note that *sambandh* markers behave exactly like adjectives in Hindi as they both modify the noun.

A similar metarule applies to the relationship between the verb and its auxiliary aspect and modality markers. The main verb itself does not expect or demand a given marker, just that there exist one. Therefore, without loss of genrality, the following metarules also stand:

$$\tau \to \tau_t$$
$$\alpha \to \alpha_a$$

Therefore the main verb of a sentence expects *some* aspect or tense marker, rather than a specific one.

5.4 Conclusion

In this chapter, the some of the most interesting syntactic properties of Hindi are explored, along with the necessary adaptations to the pregroup framework which allows these properties to be well represented. The ideas presented in this chapter, including morphological type reduction, restricting word order alternation, and subtyping analysis, are not limited to the context of Hindi syntax either. The ideas here are applicable and useful for English, as discussed already, as well as any other language which has the same, or similar, constraints in syntactic representation. For example, the discussion on the representation of clitics in Persian and Italian could be augmented with morphological type reduction in order to ascertain a consistent typing system, rather than introducing new types that overbear the calculus. Furthermore, restricting word order alternation using templatization provides a unique insight into the relationship between cnostituency grammar and the relationship between the constituents.

Fundamentally, pregroup representation aims to capture the grammaticality of a sentence in a given language by allowing adaptations to the calculus. However, by allowing restricted word order alternation to be implemented in such a manner, one of the insights provided by the reduction pattern is its similarity to a dependency parse. The relationship between constituents in templatization, as well as the type hierarchy of each word or morpheme in a given sentence, hint towards using dependency structures to better represent compositional and constituency information.

Chapter 6

Bridging the Gap: From Constituency to Dependency

Categorial and dependency grammars are two of the most popular paradigms of syntactic representation. While categorial grammars use mathematical formalism to parse constituencies with predetermined types and a predefined rule set, dependency grammars rely on parsing the relationship between individual words of a sentence from a group of relations. Most relevant literature on these paradigms treats them as unrelated schools of studying the nature of sentence structure; therefore little work has been done in a unified representation that can reduce to both. In this chapter, I show that a pregorup grammar provides an excellent basis for an abstract unified representation of syntactic structure, accounting for both constituency and dependency information. I also show that this parsing analysis is not restricted to Hindi, and can be adapted to various other languages with a uniform method to adopt relevant syntactic constraints into the framework.

6.1 Categorial and Dependency Parsing

The formal representation of natural language syntax is an age-old focus of linguistics, as a tractable representation of sentence structure can transcend the boundaries of the language itself. Two prominent abstractions to the study of syntax are phrase structure grammars, and dependency grammars. Phrase structure grammar indicates the relationship between constituents, a group of adjacent words or phrases with a single, uniform syntactic behavior [61]. These units are provided a type (such as noun, verb, or adjective phrase), and are abstracted over until the whole sentence has been provided a type as a single syntactic unit. On the other hand, dependency grammar studies the relationships between each word in a sentence from a fixed set of relations known as dependencies. Therefore, unlike phrase structure grammar which identifies contiguous units, dependency grammar does not concern itself with the word order in the sentence [75].

Since dependency grammar does not depend on contiguous constituents, it can easily be extended to free word order languages. Furthermore, since dependency parsing allocates directed relationships between corresponding syntactic roles, the idea of Universal Dependencies (UD) has been extensively researched in contemporary literature. UD identifies a large number of syntactic relations between words

which are common across languages and language families, and incorporates those into a universal lexicon of syntactic relations. While UD parses are shallow and might not contain language specific rules, the parses are still representative of the syntactic structure of the sentence, which is generally determined probabilistically.

In this chapter, I use some developments over pregroup calculus, and use it as an intermediate representation for seamless interoperability between UD parsing, and phrase structure parsing. Prior to this, some work has been done on constructing interoperability between categorial and dependency parses, such as link grammars and dependency relations in Lambek calculus. However, this is the first time a categorial grammar representation is used as an independent combined representation of both constituency, and dependency.

Therefore in this chapter, I explore three questions:

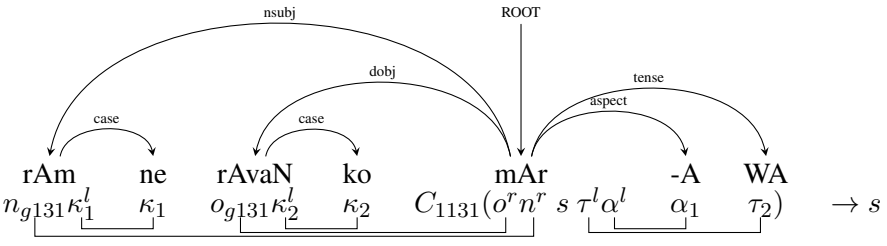
- Is there a notion of interoperability between dependency parsing and categorial grammar evaluation?
- What does this interoperability entail for the study of syntax across languages?
- What are some computational applications of this perspective?

In the following sections, I first intuit then prove the interoperability of these representations, then speak to the impact of such a representation on general syntactic notions, including constructing a viable type-set and set of relations, dealing with cross-lingual properties including restricted word order alternations and cliticization, and addressing a possible application in grammatical error correction.

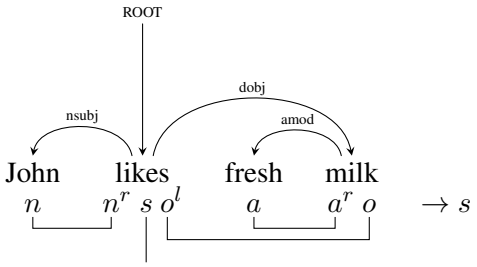
6.2 Establishing a Single Joint Representation

One of the easiest ways to showcase the interoperability between the pregroup grammar framework and dependency parses is by a diagrammatic visualization.

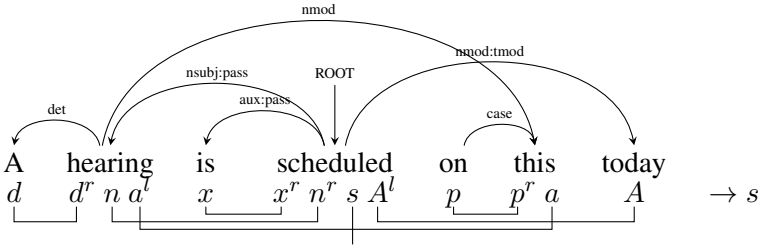
(31) raam ne raava.n ko maaraa thaa
Ram.ERG. CASE Ravan.ACC.SING. CASE kill.PERF. be.PAST.MASC.
Ram had killed Ravan



Note that this application also words for English as presented below:



The nature of the reductions above indicates that the pregroup and dependency parses of these sentences are fairly similar, so much so that the arc diagram below the example sentence is the same as the arcs of the dependency above the example sentence. Even the chosen basic types are more oriented towards relations as opposed to parts of speech. Another example, this time with non-projective dependencies, is presented below.



Non-projective dependency representations can be reduced using pregroup analyses due to the properties of precyclic reduction. The non-precyclic reduction of the above sentence would be "A hearing on this is scheduled today", which brings the nominal modifier directly next to the noun phrase. However, since the nominal modifier is a prepositional phrase, demonstrating the precyclic reduction of the overlapping arcs of the non-projective representation is trivial. This also lends insight into the nature of projectivity for dependency parsing, especially in relatively fixed word order languages. Note that the head word of the non-projective dependency still has projective dependents, which serve as its constituents.

As mentioned before, dependency parsing is a mechanism to determine the relationship between words of a sentence from a fixed set of relations. In dependency parsing, unlike the categorial models, the words are not provided types. Therefore, where categorial grammars have typed words but untyped relations, dependency grammar has untyped words but typed relations. Also note that dependencies are always directed, which means that the relation between two words is from one to the other. For example, in the sentence John likes fresh milk; John is the nominal subject of likes. This is important to note, as constituency relations are not directed.

Building upon the concept of dependency parsing, other syntactic and morphological features began to be associated with dependency parsing. While word order was not a constraint in representing dependencies, morphological and syntactic features provided insights into the structural characteristics of sentence, and therefore were features in determining the dependency grammar of a language. In order to annotate all associated features using a single annotation paradigm, universal dependencies were introduced.

Universal dependencies have been defined as: “Universal Dependencies (UD) is a framework for consistent annotation of grammar (parts of speech, morphological features, and syntactic dependencies) across different human languages” [103].

Therefore, for the sentence “Nearly 30,000 people packed the streets of San Francisco.”, the UD parse looks as follows:

1	Nearly	nearly	ADV	RB	_	2	advmod	Discourse=result:44-¿39—Entity=(person-58
2	30,000	30000	NUM	CD	NumType=Card	3	nummod	_
3	people	people	NOUN	NNS	Number=Plur	4	nsubj	Entity=person-58)
4	packed	pack	VERB	VBD	Mood=Ind—Tense=Past—VerbForm=Fin	0	root	_
5	the	the	DET	DT	Definite=Def—PronType=Art	6	det	Entity=(place-59
6	streets	street	NOUN	NNS	Number=Plur	4	obj	_
7	of	of	ADP	IN	_	8	case	_
8	San	San	PROPN	NNP	Number=Sing	6	nmod	Entity=(place-5
9	Francisco	Francisco	PROPN	NNP	Number=Sing	8	flat	Entity=place-59)place-5)

A similar example of a UD parse for the sentence

(32) mandir ke pUrva hisse mein ek swUpa hE
temple.ACC. CASE east part.LOC CASE one statue.NOM be.PRES.3
There is a statue in the eastern part of the temple.

is provided below:

1	mandir	mandir	NOUN	NN	Case=Acc|Gender=Masc|Number=Sing|Person=3	4	nmod	Vib=0_kA|Tam=0|ChunkId=NP|ChunkType=head
2	ke	kaa	ADP	PSP	AdpType=Post|Case=Acc|Gender=Masc|Number=Sing	1	case	ChunkId=NP|ChunkType=child
3	puurva	puurva	ADJ	JJ	Case=Acc	4	amod	ChunkId=NP2|ChunkType=child
4	hisse	hisse	NOUN	NN	Case=Acc|Gender=Masc|Number=Sing|Person=3	7	nmod	Vib=0_mein|Tam=0|ChunkId=NP2|ChunkType=head
5	mein	mein	ADP	PSP	AdpType=Post	4	case	ChunkId=NP2|ChunkType=child
6	ek	ek	NUM	QC	NumType=Card	7	nummod	ChunkId=NP3|ChunkType=child
7	stuupa	stuupa	NOUN	NN	Case=Nom|Gender=Masc|Number=Sing|Person=3	0	root	Vib=0|Tam=0|ChunkId=NP3|ChunkType=head
8	hai	hai	AUX	VM	Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act	7	cop	Vib=hE|Tam=hE|ChunkId=VGF|ChunkType=head |Stype=declarative

The columns provide the position of the word, the word in the sentence, the root of the word, the coarse and fine grained part-of-speech, morphological features, the position of the root, and the dependency relation. Much like the pregroup parse before, which can be universalized based on a set of basic types, the universal dependencies have established syntactic features that can represent multiple languages.

As seen in the examples presented above, the common features which stand out are the following:

- Both systems rely on a combination of parts of speech and syntactic features. For the pregroup analysis that information serves as the basic type while for dependencies it is a feature that helps determine the relation.
- Both systems of grammar have probabilistic elements. While reductions in pregroups are algorithmic, the basic type assignment itself is based on distribution more than rules. The same goes for relations in dependency parsing.
- UD and pregroup parsing are naturally extensible representations of syntactic features which explicitly or implicitly rely on morphological characteristics like case, and serve as similar descriptions of grammar.

The fundamental difference in the parses is the perspective of which element needs to be categorized, the words in the sentence, or the relationship between them. A sentence parsed using pregroup grammar forms the question: "What is the type of each word, and what types does it demand?" while the dependency parse forms the question: "What is the type of the relation between each word of the sentence?" While these are fundamentally different perspectives of sentence structure, they are extensionally (algorithmically and computationally) equivalent, so as to say: "The types which a word demands the relations between it and the other words of that sentence."

Therefore, it is possible to represent the information presented in a UD parse as pregroup formalism. This can be done by coalescing the part-of-speech and dependency relations into the parse diagram by reconstructing the basic type set, as well as extract specific morphological feature information. The representative feature information can be further constrained for establishing parsing rules for word order and constituency based grammar as well, because the relationship between basic types (despite restructuring) remains that of a constituency grammar.

Here, I note the importance of language specific metarules in order to limit basic types and transformations from being overloaded. Metarules incorporate sentence type information for successful reductions. Metarules also improve expressivity of the underlying grammar, by providing reductions which would otherwise not be allowed. For example, syntactic constraints which allow statements to become questions for the same word (such as He has seen her. to Has he seen her?). Note that in example He has seen her, the word has is an auxiliary, while in the question Has he seen her?, while it remains an auxiliary, it takes the burden of the interrogative marker.

Hence, pregroups can bridge the gap between dependency and categorial grammars by creating a framework in a language agnostic setting wherein:

- Basic types for the pregroup grammar are defined using the universal dependency and part-of-speech information,
- The morphological features extracted are used for fine-grained analysis including gender, number, person, and case agreement between different basic types, and
- Language specific metarules are constructed which ensure grammatical word order analysis with an economized basic type information.

6.3 An Analysis of Interoperability

The relationship between pregroup categorial grammar and dependency parsing can be exploited to develop intricate rule based algorithmic systems for analyzing syntax. One of the most prominent applications of this is in grammar checking to assess the syntactic well-formedness of a sentence. In this endeavor, categorial grammars are considered generalizations over syntactic structures using mathematical models, which make an algorithmic parsing implementation feasible.

In order to type-check the correctness of a sentence, the first step would be to establish the relationship the relevant parameters or features for evaluating the grammaticality of a sentence. These parameters include agreement between subject and verb (and other relations based on the language) for gender, number, person, and case; word order both within and between constituents, dangling modifiers, and incomplete sentences. Pure dependency parsing is not a good measure of grammaticality, as relations between words exist in incomplete phrases and clauses as well, and in the absence of a main verb, the head of the constituent is the root of the phrase. However, as categorial grammars model syntactic structure, whether or not a sentence reduces to the s type (the type of the sentence or the main verb) is a crucial determining factor of the grammaticality of the sentence.

Completeness of sentences is evaluated by determining whether the sentence reduces to the expected type of the sentence (such as s for statements, q for questions, etc.). This test also works for word order evaluations, such that if a given word order alternation reduces to the type of the sentence, then the sentence is well formed and well typed. Empirically, a sentence which has been parsed using a universal dependency parser, and therefore contains the part-of-speech and morphological feature analyses, can be evaluated using no more than three algorithms under the constraints of pregroup calculus. The first algorithm creates a mapping between the morphological features, parts of speech, and basic types to type individual words. The second algorithm provides the relevant adjoints in the expected syntactic order, and the third is a linear or polynomial time parsing algorithm. For directed edges in dependency analysis, the word adjointed type "demands" the words of that type.

6.4 Constructing a Grammar Checking Module

In this section, I discuss the development of the pregroup-based grammar checking module. I use a combination of four algorithms for the detection of grammatical correctness in a given sentence.

1. **Type Mapping**: The mapping algorithm consists of two parts, basic type assignment and adjoint assignment. The former is a linear time algorithm for mapping the dependency labels to given basic types using a simple hashmap. The adjoint assignment is done using the dependency label information. This generates a dictionary of words and their corresponding simple types. This algorithm has been detailed in Algorithm 1

2. **Moroz Pregroup Reduction**: This is a polynomial time dynamic parsing algorithm. [100] pregroup parsing algorithm requires a dictionary mapping of tokens (which is extracted from step 1) to simple types and reduce by incorporating language specific metarules mentioned in the sections above. If the sentence reduces to the root, I analyze the agreement rules, that is mentioned in step 3. If the sentence does NOT reduce to root, the error is the arrangement of the syntactic constituents, which is analyzed in step 4.

3. **Agreement Checking**: I convert the problem of agreement rule analysis to a type checking problem by determining the features for each basic type using the CoNLL-U features [118]. This al-

gorithm is then used to find and isolate instances of violations in agreement rules between words between a word and the reductions constructed in the Moroz parse (Algorithm 2.

4. **Word order analysis**: I also perform a test to analyze whether the word order of the sentence is correct using the language specific metarules, and the two-step reduction algorithm from chunks [54] (Algorithm 3). A common example of the need for this analysis are the adverbs or prepositional phrases in English, which are relatively free word order.

Algorithm 1 Type Mapping

```
procedure TYPE MAPPING(pos, dep, map)
    p(x) ← POS tag of word x
    d(x) ← Dep. label of word x
    deps(x) ← Children of word x
    ix(x) ← Index of word x
    map(p(x), d(x)) ← basic type of word x
    S ← sentence
    dict ← dictionary of words to simple types
    for word in S do
        P ← p(word)
        D ← d(word)
        b_type ← map(P, D)
        for child in deps(word) do
            Pc ← p(child)
            Dc ← d(child)
            if ix(child) < ix(word) then
                r_adj[child] ← map(Pc, Dc)
            else
                l_adj[child] ← map(Pc, Dc)
        dict[word] ← (r_adj, b_type, l_adj)
    return dict
```

Algorithm 2 Agreement Detection

```
procedure AGREEMENT(dep, dict, feats)
    F(x) ← Morph. features of word x
    deps(x) ← Children of word x
    S ← sentence
    for word in S do
        for child in deps(word) do
            if f in F(word) and F(child) then
                assert F(word) == F(child)
```

Algorithm 3 Word Order Checking

```
procedure WORD ORDER(dict, metarules, pos)
    p(x) ← POS tag of word x
    deps(x) ← Children of word x
    M(p) ← Metarules of POS tag p
    S ← sentence
    for word in S do
        m ← M(p(word))
        for child in deps(word) do
            Pc ← p(child)
            assert m includes Pc
```

6.5 Conclusion

In this chapter, I provided some insight into the relationship between categorial and dependency grammars. I first summarized each parsing technique by analyzing the elements in each parse, the underlying philosophy of typing words versus typing relations, and examined the differences between the two. With this in mind, I established the importance of providing encompassing and interpretable types. I provided example sentences of reductions under each paradigm, pregroups for categorial and UD for enhanced, feature-oriented dependency parsing. The examples and their diagrammatic representation showcased the similarity between the two parses, and use it to establish a notion of interoperability between the two perspectives of syntactic parsing. A computational application of this interoperability is grammatical error correction.

Chapter 7

Conclusions

This thesis has been a culmination of the idea of a tractable lexical categorial grammar representation of some of the major aspects of Hindi syntax, with a view towards compositionality. The main contributions of this thesis included:

1. A cohesive detailing of some important properties of Hindi sentence and phrase structure,

2. An analysis of word order alternation using a categorial representation using consistent mathematical theorems,

3. An exploration into the morpholexical categorial representation of some interesting properties of Hindi syntax, and

4. An overarching perspective of interoperability between dependency and categorial grammars in the pregroup framework across languages.

In this concluding chapter, I summarize each of these contributions, and point towards exciting new horizons for research in this field.

Hindi is often externally observed as a free word order language, where words can be shuffled around with few syntactic restrictions. Due to this, representations of Hindi, especially those in recent literature for computational linguistics and natural language processing, create systems where *vibhaktis* and other case markers, tense, and aspect markers are treated only as bound morphemes, regardless of the fact they are written separately from their respective heads. While the notion of a word in linguistics is heavily debated, the idea of roles is less so, and therefore such a system is also intrinsically consistent for analysis, but inconsistent on the surface. However, constructing a categorial grammar for inconsistent surface representations are not only less insightful, they do not capture the intricate relationship between syntax and nanosyntax which lexicalized surface representations with hybrid morpholexical units provide. This thesis provides a cohesive knowwledge base which can be used to extend the categorial grammar study of Hindi syntax for other parts of speech and structures such as modalities in the verb phrase, honorifics and their relevant agreement rules, etc.

One of the major contributions of this thesis was to detail this limited grammar in order to isolate some key notions about word order alternation, which extends to other languages as well. Various languages allow different degrees of freedom in shuffling the words or constituents. This degree generally depends on richness of the inflections in the language, and the importance of relative position on the role of a given word or phrase. Pregroup grammars are one of the few categorial grammars which had allowed a consistent representation of *complete* word order alternation. However, there were no restrictions on the movement of basic types, which also extends to not distinguishing between which basic types (parts of speech or syntactic roles) could be freely shuffled. Beyond extraordinarily simple fractions of some language's grammars, this property is not useful. However, in this thesis, I restructured precyclicity in pregroup calculus to introduce mechanisms to actively restrict word order alternation, either by enforcing external representations of rules on the reductions, or more elegantly, on the type system itself, an approach which has not been attempted before. The applications of such a mechanism are manifold, including the fact that both Hindi and English, languages on the ends of the word order alternation spectrum, can be easily represented using this new model with an extended grammar than before.

This leads to many interesting research directions about how well such a system can be parsed, its computational complexity, and most importantly, the extensions to the grammar that such a mechanism creates. The impact of improving representation of word order alternation and constituent shuffling on compositional sentence representations is yet another open direction of research. As mentioned in the introduction, pregroups were among the initial frameworks of compositional representation used for compoitional distributional models of meaning. Improving constituent shuffling implies that a fixed tensorial reduction of that model needs to be revisited and improved upon as well, and can also be extended to host of languages under this improved paradigm.

Most notably, while pregroup calculus is categorial and constituency-driven in both parsing and representation, it is impossible not to note the similarity of the arc-reduction diagrams to dependency parsing. Each pregroup reduction diagram identifies a root, and associates a valency for the surrounding lexical items, which mimics the idea of dependencies while maintaining the structural ideology of a phrase structure, categorial parse. There have been multiple previous attempts to link these modes of syntactic study, either by introducing combined representations such as link grammars, or probabilistic models which, given one, generate the other. However, given that the notion of a dependency grammar is notoriously hard to define, these solutions do not showcase the inherent similarity and interoperability between these two perspectives of syntax. A sentence with contiguous units have dependency relations among them, and this can be isolated at the depth of the dependency tree top-down. Similarly, the words in a constituent are likely to have a set of dependency relations among them. Combining the ability to represent word order alternation and shuffling into this, it is possible to understand how pregroups act as the interoperable syntactic representation which combines categorial and dependency perspectives. Developing parsers which utilize this information efficiently is yet another interesting direction of future work.

Related Publications

Goud, Jaipal Singh, Pranav Goel, Alok Debnath, Suhan Prabhu, and Manish Shrivastava. "A semantico-syntactic approach to event-mention detection and extraction in hindi." In Workshop on Interoperable Semantic Annotation (ISA-15), p. 63. 2019.

Debnath, Alok, and Manish Shrivastava. "A Pregroup Representation of Word Order Alternation using Hindi Syntax." Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop. 2019.

Goel, Pranav, Suhan Prabhu, Alok Debnath, and Manish Shrivastava. "Event Centric Entity Linking for Hindi News Articles: A Knowledge Graph Based Approach." In Proceedings of the 16th International Conference on Natural Language Processing, pp. 45-55. 2019.

Prabhu, Suhan, Pranav Goel, Alok Debnath, and Manish Shrivastava. "A Language Invariant Neural Method for TimeML Event Detection." In Proceedings of the 16th International Conference on Natural Language Processing, pp. 56-64. 2019.

Pant, Kartikey, Venkata Himakar Yanamandra, Alok Debnath, and Radhika Mamidi. "SmokEng: Towards Fine-grained Classification of Tobacco-related Social Media Text." In Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), pp. 181-190. 2019.

Debnath, Alok, Nikhil Pinnaparaju, Manish Shrivastava, Vasudeva Varma, and Isabelle Augenstein. "Semantic textual similarity of sentences with emojis." In Companion Proceedings of the Web Conference 2020, pp. 426-430. 2020.

Goel, Pranav, Suhan Prabhu, Alok Debnath, Priyank Modi, and Manish Shrivastava. "Hindi TimeBank: An ISO-TimeML Annotated Reference Corpus." In 16th Joint ACL-ISO Workshop on Interoperable Semantic Annotation PROCEEDINGS, pp. 13-21. 2020.

Prabhu, Suhan, Ujwal Narayan, Alok Debnath, S. Sumukh, and Manish Shrivastava. "Detection and Annotation of Events in Kannada." In 16th Joint ACL-ISO Workshop on Interoperable Semantic Anno-

tation PROCEEDINGS, pp. 88-93. 2020.

Bhat, Siddharth, Alok Debnath, Souvik Banerjee, and Manish Shrivastava. "Word Embeddings as Tuples of Feature Probabilities." In Proceedings of the 5th Workshop on Representation Learning for NLP, pp. 24-33. 2020.

Debnath, Alok, and Michael Roth. "A Computational Analysis of Vagueness in Revisions of Instructional Texts." In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pp. 30-35. 2021.

Debnath, Alok, and Manish Shrivastava. "A Categorial Grammar Perspective of Universal Dependencies." Linguistik International. 2021.

Debnath, Alok, and Manish Shrivastava. "A Computational Algebraic Analysis of Hindi Syntax." Journal of Logic, Language, and Information, Springer Publishing, 2021.

Bhattacharya, Arghya, Alok Debnath, and Manish Shrivastava. "Enhancing Aspect Extraction in Hindi." In 4th Workshop on e-Commerce and NLP, 2021.

Bibliography

[1] V. M. Abrusci. Phase semantics and sequent calculus for pure noncommutative classical linear propositional logic. *The Journal of Symbolic Logic*, 56(4):1403–1451, 1991.

[2] R. K. Agnihotri. *Hindi: An essential grammar*. Routledge, 2013.

[3] T. Ahmed, M. Butt, A. Hautli, and S. Sulger. A reference dependency bank for analyzing complex predicates. In *LREC*, 2012.

[4] K. Ajdukiewicz. Die syntaktische konnexitat. *Studia philosophica*, pages 1–27, 1935.

[5] B. R. Ambati, S. Husain, J. Nivre, and R. Sangal. On the role of morphosyntactic features in hindi dependency parsing. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 94–102. Association for Computational Linguistics, 2010.

[6] H. Andréka and S. Mikulás. Lambek calculus and its relational semantics: completeness and incompleteness. *Journal of Logic, Language and Information*, 3(1):1–37, 1994.

[7] E. Bach. Some generalizations of categorial grammars. In *The formal complexity of natural language*, pages 251–279. Springer, 1984.

[8] D. Bankova, B. Coecke, M. Lewis, and D. Marsden. Graded entailment for compositional distributional semantics. *arXiv preprint arXiv:1601.04908*, 2016.

[9] D. Bankova, B. Coecke, M. Lewis, and D. Marsden. Graded hyponymy for compositional distributional semantics. *Journal of Language Modelling*, 6(2):225–260, 2018.

[10] Y. Bar-Hillel. A quasi-arithmetical notation for syntactic description. *Language*, 29(1):47–58, 1953.

[11] Y. Bar-Hillel, G. (C.), E. Shamir, and C. Caifman. *On categorial and phrase-structure grammars*. Weizmann Science Press, 1960.

[12] D. Bargelly. Pregroup grammars and the chinese verb: What do they tell us? *Linguistic Analysis*, 36(1):49–72, 2010.

[13] G. Barry and M. Pickering. Dependency and constituency in categorial grammar. *Barry, G. and Morrill, G*, 1990.

[14] D. Béchet. Parsing pregroup grammars and lambek calculus using partial composition. *Studia logica*, 87(2-3):199–224, 2007.

[15] D. Béchet and A. Foret. A pregroup toolbox for parsing and building grammars of natural languages. *Linguistic Analysis*, 36(1):473, 2006.

[16] D. Béchet and A. Foret. Ppq: a pregroup parser using majority composition. In *Proceedings of Parsing with Categorial Grammars, ESSLLI workshop, Bordeaux, France*, 2009.

[17] A. Bharati and R. Sangal. Parsing free word order languages in the paninian framework. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 105–111, 1993.

[18] R. Bhatt, B. Narasimhan, M. Palmer, O. Rambow, D. Sharma, and F. Xia. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pages 186–189, 2009.

[19] W. Buszkowski. Lambek grammars based on pregroups. In *International Conference on Logical Aspects of Computational Linguistics*, pages 95–109. Springer, 2001.

[20] W. Buszkowski. Cut elimination for the lambek calculus of adjoints. 2002.

[21] W. Buszkowski. Sequent systems for compact bilinear logic. *Mathematical Logic Quarterly: Mathematical Logic Quarterly*, 49(5):467–474, 2003.

[22] W. Buszkowski. Lambek calculus with nonlogical axioms. *Language and Grammar. Studies in Mathematical Linguistics and Natural Language*, pages 77–93, 2005.

[23] W. Buszkowski. Lambek calculus and substructural logics. *Linguistic Analysis*, 36(1):15, 2006.

[24] W. Buszkowski, Z. Lin, and K. Moroz. Pregroup grammars with letter promotions: Complexity and context-freeness. *Journal of Computer and System Sciences*, 78(6):1899–1909, 2012.

[25] W. Buszkowski and K. Moroz. Pregroup grammars and context-free grammars. *Computational Algebraic Approaches to Natural Language, Polimetrica*, 121, 2008.

[26] M. Butt. The light verb jungle: Still hacking away. *Complex predicates: Cross-linguistic perpectives on event structure. Cambridge University Press*, 2010.

[27] M. Butt and T. H. King. The status of case. In *Clause structure in South Asian languages*, pages 153–198. Springer, 2004.

[28] K. Cardinal. *An algebraic study of Japanese grammar*. PhD thesis, McGill University Montreal, 2002.

[29] K. Cardinal. Type grammar meets japanese particles. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, pages 142–149, 2006.

[30] K. Cardinal. A pregroup analysis of japanese causatives. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation*, pages 96–104, 2007.

[31] A. Carnie. *Syntax: A generative introduction*, volume 18. John Wiley & Sons, 2012.

[32] C. Casadio. Semantic categories and the development of categorial grammars. In *Categorial grammars and natural language structures*, pages 95–123. Springer, 1988.

[33] C. Casadio. Applying pregroups to italian statements and questions. *Stud Logica*, 87(2-3):253–268, 2007.

[34] C. Casadio. Agreement and cliticization in italian: a pregroup analysis. In *International Conference on Language and Automata Theory and Applications*, pages 166–177. Springer, 2010.

[35] C. Casadio. Agreement and cliticization in italian: A pregroup analysis. In A. Dediu, H. Fernau, and C. Martín-Vide, editors, *Language and Automata Theory and Applications, 4th International Conference,*

LATA 2010, Trier, Germany, May 24-28, 2010. Proceedings, volume 6031 of *Lecture Notes in Computer Science*, pages 166–177. Springer, 2010.

[36] C. Casadio and A. Kislak-Malinowska. Italian clitic patterns in pregroup grammar: State of the art. In C. Casadio, B. Coecke, M. Moortgat, and P. Scott, editors, *Categories and Types in Logic, Language, and Physics - Essays Dedicated to Jim Lambek on the Occasion of His 90th Birthday*, volume 8222 of *Lecture Notes in Computer Science*, pages 156–171. Springer, 2014.

[37] C. Casadio and J. Lambek. An algebraic analysis of clitic pronouns in italian. In *International Conference on Logical Aspects of Computational Linguistics*, pages 110–124. Springer, 2001.

[38] C. Casadio and J. Lambek. An algebraic analysis of clitic pronouns in italian. In P. de Groote, G. Morrill, and C. Retoré, editors, *Logical Aspects of Computational Linguistics, 4th International Conference, LACL 2001, Le Croisic, France, June 27-29, 2001, Proceedings*, volume 2099 of *Lecture Notes in Computer Science*, pages 110–124. Springer, 2001.

[39] C. Casadio and J. Lambek. A tale of four grammars. *Studia Logica*, 71(3):315–329, 2002.

[40] C. Casadio and M. Sadrzadeh. Clitic movement in pregroup grammar: a cross-linguistic approach. In *International Tbilisi Symposium on Logic, Language, and Computation*, pages 197–214. Springer, 2009.

[41] C. Casadio and M. Sadrzadeh. Cyclic pregroups and natural language: a computational algebraic analysis. In *CILC*, pages 349–363. Citeseer, 2011.

[42] C. Casadio and M. Sadrzadeh. Word order alternation in sanskrit via precyclicity in pregroup grammars. In *Horizons of the Mind. A Tribute to Prakash Panangaden*, pages 229–249. Springer, 2014.

[43] C. Casadio and M. Sadrzadeh. Cyclic properties: from linear logic to pregroups. *3 SILFS*, page 139, 2016.

[44] D. Chakrabarti, H. Mandalia, R. Priya, V. Sarma, and P. Bhattacharyya. Hindi compound verbs and their automatic extraction. *Coling 2008: Companion volume: Posters*, pages 27–30, 2008.

[45] N. Chomsky. *Syntactic structures*. Walter de Gruyter, 1957.

[46] S. Clark, B. Coecke, and M. Sadrzadeh. A compositional distributional model of meaning. In *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*, pages 133–140. Oxford, 2008.

[47] B. Coecke, E. Grefenstette, and M. Sadrzadeh. Lambek vs. lambek: Functorial vector space semantics and string diagrams for lambek calculus. *Annals of pure and applied logic*, 164(11):1079–1100, 2013.

[48] B. Coecke and M. Lewis. A compositional explanation of the 'pet fish'phenomenon. In *International Symposium on Quantum Interaction*, pages 179–192. Springer, 2015.

[49] B. Coecke, M. Sadrzadeh, and S. Clark. Mathematical foundations for a compositional distributional model of meaning. *arXiv preprint arXiv:1003.4394*, 2010.

[50] B. Comrie. Reflections on verb agreement in hindi and related languages, 1984.

[51] G. D. l. Cuevas, A. Klinger, M. Lewis, and T. Netzer. Cats climb entails mammals move: preserving hyponymy in compositional distributional semantics. *arXiv preprint arXiv:2005.14134*, 2020.

[52] P. de Groote. The non-associative lambek calculus with product in polynomial time. In *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 128–139. Springer, 1999.

[53] P. De Groote. Towards abstract categorial grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 252–259, 2001.

[54] A. Debnath and M. Shrivastava. A pregroup representation of word order alternation using hindi syntax. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop (forthcoming)*, 2019.

[55] S. Degeilh. Noun phrases and postverbal clitic pronouns meet lambek's pregroups in french. 2002.

[56] S. Degeilh and A. Preller. Pregroups and the french noun phrase. 2003.

[57] S. Degeilh and A. Preller. Efficiency of pregroups and the french noun phrase. *Journal of Logic, Language and Information*, 14(4):423–444, 2005.

[58] D. Dowty. The dual analysis of adjuncts/complements in categorial grammar. *Modifying adjuncts*, 33, 2003.

[59] A. Foret. Pregroup calculus as a logic functor. In *International Workshop on Logic, Language, Information, and Computation*, pages 147–161. Springer, 2007.

[60] G. Frege. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought. *From Frege to Gödel: A source book in mathematical logic*, 1931:1–82, 1879.

[61] G. Gazdar, E. Klein, G. K. Pullum, and I. A. Sag. *Generalized phrase structure grammar*. Harvard University Press, 1985.

[62] E. D. Gutierrez, E. Shutova, T. Marghetis, and B. Bergen. Literal and metaphorical senses in compositional distributional semantic models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 183–193, 2016.

[63] J. Hedges and M. Sadrzadeh. A generalised quantifier theory of natural language in categorical compositional distributional semantics with bialgebras. *arXiv preprint arXiv:1602.01635*, 2016.

[64] E. Husserl. *Ideas: General introduction to pure phenomenology*. Routledge, 2012.

[65] A. K. Joshi and Y. Schabes. Tree-adjoining grammars. In *Handbook of formal languages*, pages 69–123. Springer, 1997.

[66] D. Jurafsky and J. H. Martin. *Speech & language processing*. Pearson Education India, 2000.

[67] Y. Kachru. Causative sentences in hindi revisited. 1971.

[68] Y. Kachru. *Aspects of Hindi grammar*. South Asia Books, 1980.

[69] Y. Kachru. *Hindi*, volume 12. John Benjamins Publishing, 2006.

[70] D. Kartsaklis. Coordination in categorical compositional distributional semantics. *arXiv preprint arXiv:1606.01515*, 2016.

[71] S. H. Kellogg and T. G. Bailey. A grammar of the hindi language: in which are treated the high hindí, braj, and the eastern hindí of the rámáyan of tulsí dás, also the colloquial dialects of rájputáná, kumáon,

avadh, ríwá, bhojpúr, magadha, maithila, etc., with copious philological notes/by sh kellogg; with notes on pronunciation by t. grahame bailey. 1972.

[72] A. Kiślak-Malinowska. Extended pregroup grammars applied to natural languages. *Logic and Logical Philosophy*, 21(3):229–252, 2012.

[73] A. B. Kleiman. Some aspects of the causative construction in hindi. 1971.

[74] O. N. Koul. *Modern Hindi Grammar*. Dunwoody Press Springfield, USA, 2008.

[75] S. Kübler, R. McDonald, and J. Nivre. Dependency parsing. *Synthesis lectures on human language technologies*, 1(1):1–127, 2009.

[76] J. Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):154–170, 1958.

[77] J. Lambek. On the calculus of syntactic types. *Structure of language and its mathematical aspects*, 12:166–178, 1961.

[78] J. Lambek. Type grammar revisited. In *International conference on logical aspects of computational linguistics*, pages 1–27. Springer, 1997.

[79] J. Lambek. Type grammar revisited. *Lecture notes in computer science*, pages 1–27, 1999.

[80] J. Lambek. Type grammar meets german word order. *Theoretical Linguistics*, 26(1-2):19–30, 2000.

[81] J. Lambek. Type grammars as pregroups. *Grammars*, 4(1):21–39, 2001.

[82] J. Lambek. A computational algebraic approach to english grammar. *Syntax*, 7(2):128–147, 2004.

[83] J. Lambek. From word to sentence: a pregroup analysis of the object pronoun who (m). *Journal of Logic, Language and Information*, 16(3):303–323, 2007.

[84] J. Lambek. Invisible endings of english adjectives and nouns. *Linguistic Analysis*, 2007.

[85] J. Lambek. *From Word to Sentence: a computational algebraic approach to grammar*. Polimetrica sas, 2008.

[86] J. Lambek. Pregroup grammars and chomsky's earliest examples. *Journal of Logic, Language and Information*, 17(2):141–160, 2008.

[87] J. Lambek. Exploring feature agreement in french with parallel pregroup computations. *Journal of Logic, Language and Information*, 19(1):75, 2010.

[88] J. Lambek. Logic and grammar. *Studia Logica*, 100(4):667–681, 2012.

[89] J. Lambek and A. Preller. An algebraic approach to the german sentence. In *Fields Summer School in Logic*, 2003.

[90] J. Lambek and A. Preller. An algebraic approach to the german sentence. *Linguistic Analysis*, 31(3/4):17–37, 2004.

[91] Y. Le Nir. From proof trees in lambek calculus to ajdukiewicz bar-hillel elimination binary trees. *Research on Language and Computation*, 1(3-4):181–201, 2003.

[92] A. Lecomte and C. Retoré. Pomset logic as an alternative categorial grammar. In *Formal Grammar*, pages 181–196. Barcelona, 1995.

[93] E. Manetta. Verb-phrase ellipsis and complex predicates in hindi-urdu. *Natural Language & Linguistic Theory*, 37(3):915–953, 2019.

[94] C. P. Masica. *The indo-aryan languages*. Cambridge University Press, 1993.

[95] J. D. McCawley. *The syntactic phenomena of English*. University of Chicago Press, 1998.

[96] R. S. McGregor. *Outline of Hindi Grammar: with exercises*. Oxford University Press, 1972.

[97] A. Montaut. The evolution of the tense-aspect system in hindi/urdu. In *Proceedings of LFG Conference 2006*, page prépublication. en ligne (CSLI publications), 2006.

[98] M. Moortgat. *Categorial investigations: Logical and linguistic aspects of the Lambek calculus*, volume 9. Walter de Gruyter GmbH & Co KG, 2020.

[99] R. Moot. A type-logical treebank for french. *Journal of Language Modelling*, 3(1):229–264, 2015.

[100] K. Moroz. Parsing pregroup grammars in polynomial time. In *2009 International Multiconference on Computer Science and Information Technology*, pages 257–264. IEEE, 2009.

[101] K. Moroz. A savateev-style parsing algorithm for pregroup grammars. In *International Conference on Formal Grammar*, pages 133–149. Springer, 2009.

[102] K. Moroz. A dynamic parsing algorithm for pregroup grammars. In *Young Researchers Forum*, page 1, 2011.

[103] J. Nivre, M.-C. De Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, et al. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, 2016.

[104] M. Palmer, R. Bhatt, B. Narasimhan, O. Rambow, D. M. Sharma, and F. Xia. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17, 2009.

[105] L. Pannitto and A. Lenci. Event knowledge in compositional distributional semantics. *IJCoL. Italian Journal of Computational Linguistics*, 5(5-1):73–88, 2019.

[106] M. Pickering and G. Barry. Dependency categorial grammar and coordination. *Linguistics*, 31(5):855–902, 1993.

[107] A. Preller. Linear processing with pregroups. *Studia Logica*, 87(2-3):171–197, 2007.

[108] A. Preller. Semantic pregroup grammars handle long distance dependencies in french. In *TALN'07: Traitement Automatique des Langues Naturelles-Atelier" Formalismes Syntaxiques de Haut Niveau"*, volume 14, pages 503–512. ATALA, 2007.

[109] A. Preller. Toward discourse representation via pregroup grammars. *Journal of Logic, Language and Information*, 16(2):173–194, 2007.

[110] A. Preller. Polynomial pregroup grammars parse context sensitive languages. *Linguistic Analysis*, 36(Lambek Festschrift):483–516, 2010.

[111] A. Preller and J. Lambek. Free compact 2-categories. *Mathematical Structures in Computer Science*, 17(doi: 10.1017/S0960129506005901):309–340, 2007.

[112] A. Preller and V. Prince. Pregroup grammars with linear parsing of the french verb phrase, 2008.

[113] M. Sadrzadeh. Pregroup analysis of persian sentences. 2007.

[114] M. Sadrzadeh. An adventure into hungarian word order with cyclic pregroups. *Models, Logics, and Higher-Dimensional Categories, a tribute to the work of Michael Makkai, Centre de Recherches Mathématiques. Proceedings and Lecture Notes*, 53:263–275, 2011.

[115] Y. Savateev. Product-free lambek calculus is np-complete. In *International Symposium on Logical Foundations of Computer Science*, pages 380–394. Springer, 2009.

[116] Y. Savateev. Product-free lambek calculus is np-complete. *Annals of pure and applied logic*, 163(7):775–788, 2012.

[117] G. Schneider. A linguistic comparison of constituency, dependency and link grammar. *Master's thesis, University of Zurich*, 1998.

[118] S. Schuster and C. D. Manning. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2371–2378, 2016.

[119] G. Singh and D. Lobiyal. A computational grammar for hindi verb phrase. In *Proceedings of International Conference on Expert Systems for Development*, pages 244–249. IEEE, 1994.

[120] S. Singh and V. M. Sarma. Verbal inflection in hindi: A distributed morphology approach. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, 2011.

[121] A. Spencer, M. Butt, and T. H. King. Case in hindi. In *Proceedings of LFG*, volume 5, pages 429–446, 2005.

[122] E. P. Stabler. Tupled pregroup grammars. *Computational and Algebraic Approaches to Natural Language, edited by Claudia Casadio and Joachim Lambek. Milano, Italia: Polimetrica*, 2008.

[123] M. Steedman. Categorial grammar. *Lingua*, 90(3):221–258, 1993.

[124] M. Steedman and J. Baldridge. Combinatory categorial grammar. *Non-Transformational Syntax: Formal and explicit models of grammar*, pages 181–224, 2011.

[125] A. Vaidya, O. Rambow, and M. Palmer. Light verb constructions with 'do'and 'be'in hindi: A tag analysis. In *Proceedings of Workshop on Lexical and Grammatical Resources for Language Processing*, pages 127–136, 2014.

[126] K. Vijay-Shanker and D. J. Weir. The equivalence of four extensions of context-free grammars. *Mathematical systems theory*, 27(6):511–546, 1994.

[127] G. Wijnholds. Categorical foundations for extended compositional distributional models of meaning. 2015.

[128] D. N. Yetter. Quantales and (noncommutative) linear logic. *The Journal of Symbolic Logic*, 55(1):41–64, 1990.