A Pregroup Representation of Word Order Alternation using Hindi Syntax

by

Alok Debnath, Manish Shrivastava

in

2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (*NAACL-2019*)
: 125
-135

Minneapolis, Minnesota, USA

Report No: IIIT/TR/2019/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2019

A Pregroup Representation of Word Order Alternation using Hindi Syntax

Alok Debnath and **Manish Shrivastava**
Language Technologies Research Center (LTRC)
Kohli Center on Intelligent Systems
International Institute of Information Technology, Hyderabad
alok.debnath@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

Pregroup calculus has been used for the representation of free word order languages (Sanskrit and Hungarian), using a construction called precyclicity. However, restricted word order alternation has not been handled before. This paper aims at introducing and formally expressing three methods of representing word order alternation in the pregroup representation of any language. This paper describes the word order alternation patterns of Hindi, and creates a basic pregroup representation for the language. In doing so, the shortcoming of correct reductions for ungrammatical sentences due to the current apparatus is highlighted, and the aforementioned methods are invoked for a grammatically accurate representation of restricted word order alternation. The replicability of these methods is explained in the representation of adverbs and prepositional phrases in English.

1 Introduction

Categorial grammars are one of the frameworks for the representation of syntactic structures of languages (Oehrle et al., 2012). A foundational problem in such formalisms, including the well established lexical formalism combinatory categorial grammars (CCG), is the representation of free word order in light of syntactic or semantic constraints presented by the language. Extensive resources following the different formalisms, such as CCG Banks (Hockenmaier and Steedman, 2007), have been developed. Development in pregroup calculus, however, has been more focused on developing formal constraints in the calculus for the representation of syntactic phenomena. In that vein, this paper aims at presenting the problem of restricted word alternation (constituent scrambling) by using the example of Hindi syntax, and uses that to develop three formal approaches to represent word alternation in the pregroup calculus framework.

Pregroups are mathematical structures which were developed initially by Lambek and can be used to analyze sentences in English algebraically (Lambek, 1997). Pregroup calculus was a revision of Lambek's previous categorial grammar called Syntactic Calculus (Lambek, 1958). Various languages have since adopted the use of pregroup calculus as the formal representation of syntax of a fragment or a particular property of the language, including Arabic (Bargelli and Lambek, 2001b), French (Bargelli and Lambek, 2001a), German (Lambek, 2000), Japanese (Cardinal, 2002), Persian (Sadrzadeh, 2007), Polish (Kislak-Malinowska, 2008), and Sanskrit (Casadio and Sadrzadeh, 2014), among others. Coecke et al. (2010)'s compositional distributional model of meaning also uses pregroup calculus as the compositional theory for grammatical types.

In the study of free word order languages (or languages with clitics), such as Italian (Casadio, 2010), Latin (Casadio and Lambek, 2005) and Hungarian (Sadrzadeh, 2011), a transformation was introduced, knows as the precyclic transformation (section 3.1). This transformation was also used for Sanskrit (Casadio and Sadrzadeh, 2014), where generation of ungrammatical sentences was restricted by disallowing certain transformations.

In this paper, we first establish a preliminary pregroup grammar of Hindi and highlight the syntactic constraints of the language. We then show the shortcomings of the current word order alternation mechanism. We also formally define the

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 125–135
Minneapolis, Minnesota, June 3 - 5, 2019.

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

Table 1: Case/Role Marking in Hindi

aforementioned restrictions presented for Sanskrit, and present two other novel methods for representing restricted word order alternation, which can be applied to other languages as well, evidenced by the examples of prepositional phrases in English.

2 Properties of Hindi Syntax

This section details the syntactic properties of Hindi, which include postpositional case marking (*karaka* and *sambandha* markers) in noun phrases, lexicalized tense and aspect markers in verb phrases, and the default word order and constituent scrambling. These properties are essential to developing a pregroup representation of Hindi syntax. Section 2.1 highlights the word order movement properties of Hindi.

In the Paninian linguistic tradition, noun phrases in Hindi have a system of case marking known as the *karaka* system. *karaka* is analogous to a case, which is marked by a *vibhakti* (case marker). The *karaka* is a syntacto-semantic system which distinctly identifies the role of a noun to a verb (Pedersen et al., 2004). Table 1 shows the *karakas*, their respective markers and their analogous cases.

The genitive case (called *sambandh*) in Hindi is not a *karaka*, as it shows the relationship of one noun to another noun. These are gender marked, to reflect the gender of the following noun. For instance the phrase "Neha's ball" will be translated as *Neha* ***kI*** *gend*.

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" (Ahmed et al., 2012). 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) (Spencer et al., 2005; Chakrabarti et al., 2008).

2.1 Restricted Word Order Alternation

Hindi follows a general SOV word order (more specifically, S-IO-DO-V) (Seddah et al., 2010).[1] In the default word order, Hindi is a head-final with a relatively free word order (Patil et al., 2008). However, constituents are often "mixed up", which may be done for focus or emphasis, but this is not always the case (Butt and King, 1996; Kidwai, 2000).

We take an example of the sentence: *raam ne sitaa ko kitaab dii* Ram erg. Sita dat. book gave-fem. "Ram gave the book to Sita" to explain the possible word orders (Ambati et al., 2010).

- Default word order (S-IO-DO-V) used above
- S-DO-IO-V: *raam ne kitaab sitaa ko dii*
- IO-S-DO-V: *sitaa ko raam ne kitaab dii*
- IO-DO-S-V: *sitaa ko raam ne kitaab dii*
- DO-S-IO-V: *kitaab raam ne sitaa ko dii*
- DO-IO-S-V: *kitaab sitaa ko raam ne dii*

Note that due to its isolating nature, the word order in the constituents remains intact, which is difficult to represent. The pregroup representation should allow only for valid constituent alternation while keeping the word order in the constituents in agreement with the grammar. The example provided in Section 4.3 explains this in detail.

3 Pregroup Calculus and Precyclicity

In this section, we define pregroups and pregroup grammars. We also explore the mathematical apparatus required to define word order change, based upon the concept of precyclicity.

A pregroup is a partially ordered monoid [2] $(P, \cdot, 1, \rightarrow, (-)^l, (-)^r)$, which has two unary operators, the left and the right adjoint such that $\forall x \in P$:

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

[1] S = subject, O = object, IO = indirect object, DO = direct object, V = verb

[2] A monoid is a set closed under an associative binary operation. Partial order indicates that the binary relation has to be reflexive, anti0symmetric and transitive.

where 1 is the identity element, the $\cdot$ operator is a concatenation operator (usually not explicitly mentioned) and the $\rightarrow$ operator indicates partial order.

Some other properties of pregroups include:

$$1^l = 1 = 1^r$$
$$(a \cdot b)^l = b^l \cdot a^l \quad (a \cdot b)^r = b^r \cdot a^r$$
$$(a^l)^l = a^{ll} \quad (a^r)^r = a^{rr}$$
$$a^{rl} = a = a^{lr}$$

Adjoints are switching in nature, which means that:

$$a \rightarrow b \implies b^l \rightarrow a^l \quad a \rightarrow b \implies b^r \rightarrow a^r$$

The operations $x^l \cdot x \rightarrow 1$ and $x \cdot x^r \rightarrow 1$ are called contractions and the operations $1 \rightarrow x \cdot x^l$ and $1 \rightarrow x^r \cdot x$ are called expansions. The monoids used for pregroup grammars are called free pregroups, which have the property that without loss of generality, contractions precede expansions. This is called the *switching lemma* (Lambek, 1999).

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 a concatenation of simple types. Pregroup grammars, much like other categorial grammars, assigns compound types to words in a sentence. A sentence is considered grammatical if it reduces (by concatenation with adjoints) to the simple type of the main verb in the sentence.

Therefore, in English a simple transitive verb will be represented as follows:

subject verb object

$$n \quad n^r \; s \; o^l \quad o \qquad \rightarrow s$$

The reductions are shown by the arcs, and the sentence reduces to type s, the type of the main verb.

A pregroup grammar is a quintuple $G = (\Sigma, P, \rightarrow, s, I)$ such that Σ is a nonempty, finite alphabet, $(P, \rightarrow)$ is a finite poset, $s \in P$, and I is a finite relation between symbols from Σ and non-empty types (on P) (Buszkowski, 2001). 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 $\mathcal{B}$.

3.1 Precyclicity in 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 (Casadio and Lambek, 2002). Since pregroups used for language formalism are free, proper pregroups (Buszkowski, 2001), the classical definition of cyclicity $a \cdot b \rightarrow c \implies b \cdot a \rightarrow c$ does not hold. However, a weak form of cyclicity, called precyclicity, is admitted, which has the following properties (Yetter, 1990):

$$pq \rightarrow r \implies q \rightarrow p^r r \tag{1}$$
$$q \rightarrow rp \implies qp^r \rightarrow r \tag{2}$$
$$pq \rightarrow r \implies q \rightarrow rp^l \tag{3}$$
$$q \rightarrow pr \implies p^l q \rightarrow r \tag{4}$$

Due to this, we obtain the following rules for precyclicity with double adjoints (Abrusci, 1991):

$$1 \rightarrow ab \overset{ll}{\Rightarrow} 1 \rightarrow ba^{ll} \tag{5}$$
$$1 \rightarrow ab \overset{rr}{\Rightarrow} 1 \rightarrow b^{rr}a \tag{6}$$

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

$$(ll) - transformation$$
$$A \rightarrow B(ab)C \rightsquigarrow^{ll} A \rightarrow B(ba^{ll})C \tag{7}$$
$$(rr) - transformation$$
$$A \rightarrow B(ab)C \rightsquigarrow^{rr} A \rightarrow B(b^{rr}a)C \tag{8}$$

These precyclic transformations provide the following two equations,

$$p^r q \leq qp^l \tag{9}$$
$$qp^l \leq p^r q \tag{10}$$

which can be seen to be empirically derived in clitic movement patterns in other languages, explored in Casadio and Sadrzadeh (2009).

4 A Preliminary Pregroup Grammar for Hindi

This section identifies the basic types and compound types used in Hindi. The basic types are pregroup pregroup reprensentations of syntactic features discussed in Section 2.

4.1 Basic Types

The basic types will be similar to the set of basic types chosen for English, $\{\pi, o, p, n, s\}$, which are personal pronouns, the direct object of a transitive verb, simple predicate, noun phrase, and sentence respectively (Lambek, 2004). The basic types for the lexicalized case markers as well as the tense and aspect markers have to be included, explained below.

The basic type κ_i and ρ represent *karaka* and *sambandh* markers (for the genitive case) respectively. The subscript on κ denotes the case of the noun that precedes it (refer Table 2). A simple example of the genitive case interaction, for the noun phrase "Ram's brother" (*raam kaa bhai*), is as:

$$(n)(n^r\rho)(\rho^r n) \rightarrow n$$

The type assignment seems to imply that in the genitive case marker is the headword of the noun phrase *raam kaa*, which is not the case. Genitives in Hindi are gender marked, and they agree with the gender of the following noun phrase, which is preserved by the current type assignment. The type assignment is reflective of the Paninian framework of modifier-modified relationship, in which the genitive case marker reflects the modification of the following noun phrase (Bharati and Sangal, 1993).

The VP consists of a verb, an optional auxiliary (denoted by α) and a tense marker (denoted by τ), in that order of occurrence. Verb transitivity does affect the verb typing, but only in the sentence. All statement verbs are given the type s. Verbs in Hindi are unique for their tense marking system, which include a gender-marked past tense marker, a gender-neutral present tense marker, and a suffixed future tense marker.

Given the semantic nature of *karaka* markers in Hindi, verb transitivity raises ambiguous cases. For example, the sentence *Ram ne seb* ***ko*** *khaaya* and *Ram ne seb khaaya* both translate to "Ram ate an apple". In this analysis, there are three distinct cases of the use of *karaka* intransitive verbs are: (1) no case markers, (2) ergative case marker on the subject and (3) ergative case marker on the subject and accusative case marker on the direct object (Palmer et al., 2009).

κ_i	*karaka*
κ_1	*karta*
κ_2	*karma*
κ_3	*karan*
κ_4	*sampradan*
κ_5	*apaadan*
κ_6	*adhikaran*

Table 2: Type given to *karaka*

4.2 Examples of Hindi Sentences

Given the set of basic types $\{\pi, s, p, o, n, \kappa_i, \rho, \alpha, \tau\}$, simple sentences can be typed in Hindi as follows. The toy examples chosen here are similar in to a few of the simple sentences of the Hindi treebank (Bhatt et al., 2009).

1. I go to school.

mein	*skool*	*jaataa*	*hun*
I	school	go-perf.-masc.	am
π	o	$o^r\pi^r s\tau^l$	τ

$$(\pi)(o)(o^r\pi^r s\tau^l)(\tau) \rightarrow s$$

2. Tina sang a song.

tinaaa	*ne*	*gaanaa*	*gaayaa*
Tina	erg.	song	sang-masc.
$n\kappa_1^l$	κ_1	o	$o^r n^r s$

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

3. Ram had hit the ball with a bat.

raam	*ne*	*gend*	*ko*
Ram	erg.	ball	acc.
$n\kappa_1^l$	κ_1	$o\kappa_2^l$	κ_2

balle	*se*	*maaraa*	*thaa*
bat	with	hit-masc.	was-masc.
$p\kappa_3^l$	κ_3	$p^r o^r n^r s\tau^l$	τ

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

$$s\,\tau^l)(\,\tau) \rightarrow s$$

4.3 Consequences of Restricted Word Order Alternation

In order to apply precyclicity rules,[3] the example chosen is a simple transitive verb from the sentence "Tina sang a song", which has been typed above in section 4.2. As above, the arcs denote a reduction, while the underline shows the arguments of the precyclic transformation.

4.

gaanaa	*tina*	*ne*	*gaayaa*
song	Tina	erg.	sang-masc.

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

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

balle	*se*	*gend*	*ko*
bat	with	ball	acc.

5.

raam	*ne*	*maaraa*	*thaa*
Ram	erg.	hit-masc.	was-masc.

$$
\begin{aligned}
&(p\kappa_3{}^l)(\kappa_3)(o\kappa_2{}^l)(\kappa_2)(n\kappa_1{}^l)(\kappa_1)(p^r o^r n^r s\\
&\quad \tau^l)(\tau)\\
&\rightarrow (p)(o)(n)(p^r \underline{o^r n^r s})\\
&\rightsquigarrow^{ll} (p)(o)(n)(\underline{p^r n^r o^l s})\\
&\rightsquigarrow^{ll} (p)(o)(n)\overline{(n^r p^l} o^l s)\\
&\rightarrow (p)(o)\overline{(p^l o^l s)}\\
&\rightsquigarrow^{rr} (p)(o)\overline{(o^r p^l s)}\\
&\rightarrow \underline{(p)(p^l s)}\\
&\rightsquigarrow^{rr} \overline{(p^l s)^{rr}(p)}\\
&\rightarrow \underline{(p^r s^{rr})}(p)\\
&\rightsquigarrow^{ll} (s^{rr} p^l)(p)\\
&\rightarrow s^{rr}\\
&\rightsquigarrow^{ll} s
\end{aligned}
$$

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:

6. *

	tinaa	*gaanaa*	*ne*	*gaayaa*
	Tina	song	erg.	sang
	$n\kappa_1^l$	o	κ_1	$o^r n^r s$

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

5 Restricting Word Movement

As seen in section 4, 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.

5.1 Pregroup Grammar Rules

Pregroup grammar rules for restricting word order movement have been briefly discussed in (Casadio, 2004) and (Casadio and Sadrzadeh, 2014). Here, instead of treating restrictive rules as an exception, we treat it as a part of the pregroup framework, by creating a formal representation of these restrictions.

For example, a word order rule in Hindi syntax is "The alternation of ANY phrase with a *karaka* marker is disallowed" (Bopche et al., 2012), it may be represented in the following way:

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

where, as discussed in Section 3, $\mathcal{B}$ is the set of all basic types in the language. A similar set of

[3] The procedure for applying the rules has been as described by Casadio and Sadrzadeh (2014) for Sanskrit.

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.

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

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

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

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

5.2 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 (Refer to the Appendix for the proof).

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:

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

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

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

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.[4]

5.3 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 3 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 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.

[4]Refer to Appendix for mathematical correctness of selective transformation.

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 3: Constituent Profiles for Hindi

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 4.2 *tinaa gaanaa ne gaayaa*. 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

6 Word Order Alternation in English

Lambek's work in English type grammar (Lambek, 2004) and the work that has followed (Preller, 2007; Stabler, 2008) 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. Therefore, the sample sentence, in the default word order, can be typed as follows ("the fields" has been typed p):

7.

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

$$(\pi)(A)(A^r\pi^r s\rho^l)(\rho p^l)(p) \rightarrow s$$

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:

8.

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

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

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

9. $*$

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

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

$$\begin{gathered}\rightarrow (A)(\pi)(\underline{A^r\pi^r s})\\ \rightsquigarrow^{ll} (A)(\pi)(\pi^r A^l s)\\ \rightarrow (A)(A^l s)\\ \rightsquigarrow^{rr} \underline{(A^r s^{rr})(A)}\\ \rightsquigarrow^{ll} (s^{rr}A^l)(A) \rightsquigarrow^{ll} s\end{gathered}$$

Therefore, the methods discussed in section 5 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.

7 Conclusion

This paper is an attempt to formalize the syntactic notion of word order alternation using pregroup grammars by expanding on the developments in the current literature and proposing two novel methods of representing restrictions in word movement. The inspiration for this development stems from the syntactic structure of Hindi, an isolating free word order language which allows only for constituent scrambling, keeping the order of the words in the constituents the same.

In order to express the syntactic properties of Hindi, a preliminary pregroup grammar for Hindi has also been established. This pregroup grammar has been developed in a manner similar to their development in other languages, with the establishment of basic types and examples of sentences represented using the grammar. Further work can be done in the development of the pregroup grammar of Hindi, in modeling agreement rules and other aspects of the Hindi syntax.

The methods proposed include the expansion and formal representation of the method used for Sanskrit, as well as two novel approaches which are selective transformation and two-step reduction. Using examples in the paper, these methods have been applied to prove their effectiveness,

and an example has also been taken from English, which does the same.

Future work in this direction could include development of pregroup representations of common cross-lingual syntactic phenomena, such as agglutenation, that would make the adaptation of pregroups to other languages easier. A more thorough grammar, based on other properties such as subject-verb agreement, verb complements and so on are other directions of developing the work.

Acknowledgements We sincerely thank the inputs and suggestions of Dr. Marie-Catherine de Marneffe. We would also like to acknowledge and thank the anonymous reviewers for their time and effort.

References

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

Tafseer Ahmed, Miriam Butt, Annette Hautli, and Sebastian Sulger. 2012. A reference dependency bank for analyzing complex predicates. In *LREC*.

Bharat Ram Ambati, Samar Husain, Joakim Nivre, and Rajeev Sangal. 2010. 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.

Daniele Bargelli and Joachim Lambek. 2001a. An algebraic approach to French sentence structure. In *International Conference on Logical Aspects of Computational Linguistics*, pages 62–78. Springer.

Donna Bargelli and Joachim Lambek. 2001b. An algebraic approach to Arabic sentence structure. *Linguistic Analysis*, 31(3):301–315.

Akshar Bharati and Rajeev Sangal. 1993. Parsing free word order languages in the Paninian framework. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 105–111. Association for Computational Linguistics.

Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pages 186–189.

Lata Bopche, Gauri Dhopavkar, and Manali Kshirsagar. 2012. Grammar checking system using rule based morphological process for an indian language. In *Global Trends in Information Systems and Software Applications*, pages 524–531. Springer.

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

Wojciech Buszkowski. 2002. Cut elimination for the Lambek calculus of adjoints.

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

Miriam Butt and Tracy Holloway King. 1996. Structural topic and focus without movement. In *Proceedings of the First LFG Conference*. Stanford: CSLI Publications.

Kumi Cardinal. 2002. *An algebraic study of Japanese grammar*. Ph.D. thesis, McGill University Montreal.

Claudia Casadio. 2004. Pregroup grammar. theory and applications.

Claudia Casadio. 2010. Agreement and cliticization in Italian: a pregroup analysis. In *International Conference on Language and Automata Theory and Applications*, pages 166–177. Springer.

Claudia Casadio and Jim Lambek. 2005. A computational algebraic approach to latin grammar. *Research on Language and Computation*, 3(1):45.

Claudia Casadio and Joachim Lambek. 2002. A tale of four grammars. *Studia Logica*, 71(3):315–329.

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

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

Debasri Chakrabarti, Hemang Mandalia, Ritwik Priya, Vaijayanthi Sarma, and Pushpak Bhattacharyya. 2008. Hindi compound verbs and their automatic extraction. *Coling 2008: Companion volume: Posters*, pages 27–30.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *arXiv preprint arXiv:1003.4394*.

Julia Hockenmaier and Mark Steedman. 2007. Ccgbank: a corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396.

Ayesha Kidwai. 2000. *XP-adjunction in Universal Grammar: Scrambling and binding in Hindi-Urdu*. Oxford University Press on Demand.

Aleksandra Kislak-Malinowska. 2008. Polish language in terms of pregroups. *Recent Computational Algebraic Approaches to Morphology and Syntax. Polimetrica, Milan*, pages 145–172.

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

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

Joachim Lambek. 1997. Type grammar revisited. In *International Conference on Logical Aspects of Computational Linguistics*, pages 1–27. Springer.

Joachim Lambek. 2000. Type grammar meets German word order. *Theoretical Linguistics*, 26(1-2):19–30.

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

Richard T Oehrle, Emmon Bach, and Deirdre Wheeler. 2012. *Categorial grammars and natural language structures*, volume 32. Springer Science & Business Media.

Martha Palmer, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17.

Umesh Patil, Gerrit Kentner, Anja Gollrad, Frank Kügler, Caroline Féry, and Shravan Vasishth. 2008. Focus, word order and intonation in Hindi. *Journal of South Asian Linguistics*, 1(1).

Mark Pedersen, Domenyk Eades, Samir K Amin, and Lakshmi Prakash. 2004. Relative clauses in Hindi and Arabic: A Paninian dependency grammar analysis. In *Proceedings of the Workshop on Recent Advances in Dependency Grammar*.

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

Mehrnoosh Sadrzadeh. 2007. Pregroup analysis of Persian sentences.

Mehrnoosh Sadrzadeh. 2011. 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.

Djamé Seddah, Sandra Koebler, and Reut Tsarfaty. 2010. Proceedings of the NAACL HLT 2010 first workshop on statistical parsing of morphologically-rich languages. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*.

Andrew Spencer, Miriam Butt, and Tracy Holloway King. 2005. Case in Hindi. In *Proceedings of LFG*, volume 5, pages 429–446.

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

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

A Appendix: The Mathematics of Selective Transformation

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 Yetter (1990), for relaxing the conditions on the cut theorem on one sided sequent calculus for pure non-commutative classical linear propositional logic. The original rules of precyclic transformation:

$$\frac{\vdash \Gamma, A}{\vdash A^{\perp\perp}, \Gamma} \qquad \frac{\vdash A, \Gamma}{\vdash \Gamma, {}^{\perp\perp}A}$$

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:

$$\frac{\vdash \Gamma_1, A, \Gamma_2 \;\; \vdash \Delta_1, A^{\perp}, \Delta_2}{\vdash \Delta_1, \Gamma_1, \Delta_2, \Gamma_2} \qquad \frac{\vdash \Gamma_1, {}^{\perp}A, \Gamma_2 \;\; \vdash \Delta_1, A, \Delta_2}{\vdash \Delta_1, \Gamma_1, \Delta_2, \Gamma_2}$$

if $\Delta_1 = \varnothing$ or $\Gamma_2 = \varnothing$. And the cut theorem in SPNCL', due to these rules, was reduced to:

$$\frac{\vdash \Gamma, A \;\; \vdash \Delta, A^{\perp}}{\vdash \Delta, \Gamma} \qquad \frac{\vdash \Gamma, {}^{\perp}A \;\; \vdash \Delta, A}{\vdash \Delta, \Gamma}$$

Precyclic pregroups are a result of a bilinear mapping of this reduction in SPNCL' to pregroups, using the interpretation map of compact

bilinear logic (Buszkowski, 2003). However, note that this reduction was made to relax the constraints on cut theorem in SPNCL. Buszkowski (2002) 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 $(-)^{\perp\perp}$ cannot be used for formulae of SPNCL. Hence, given the compact map from bi-linear logic to pregroups, the analogous ll- and rr-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.