Journal Impact Factor (JIF)™ 2022: 0.2 | 5 Year: 0.2 | *Source: Journal Citation Report™ 2023 (Clarivate Analytics)*

RUPKATHA JOURNAL
ON INTERDISCIPLINARY STUDIES IN HUMANITIES

AESTHETIX MEDIA

ISSN 0975-2935 | Indexed in the Web of Science Core Collection™ Emerging Sources Citation Index (ESCI)

https://doi.org/10.21659/rupkatha.v16n2.01 | Volume 16, Number 2, 2024 | Dynamic Impact Dimensions

Research Article

Semantic Model for Fragment of Hindi (Part 2)

[Access Part 1 of the article ↗]

Vivek Tripathi[1]* & Dinesh Rathod[2]

[1]Research Scholar, Indian Institute of Technology (BHU) Varanasi. *Corresponding author.

[2]Research Scholar, Indian Institute of Science, Bangalore.

Abstract

This paper proposes a formal model for semantic analysis of a fragment of the Hindi language. This paper uses referential noun phrases, transitive and intransitive verb phrases and logical constants to compute the meaning of its sentences generated from the Hindi part-of-speech-tagged corpus features. The paper presents cases of conjunction and negation enriched with idempotent laws that provide semantic computation of simple and complex well-formed formulas. Our system works for any model, with one such model described in our glossary. It deals with the set-theoretic study of essential syntactic categories of Hindi, suggesting the suitability of our rule-based syntactic arrangement and model-based semantic computation by implementing them through an in-house software tool.

Keywords: Natural Languages Processing. Hindi Language Processing. Parser. Context-Free Grammar. Hindi Semantics. Semantic Model for Hindi. Montague Grammar.

Conflicts of Interest: The authors declared no conflicts of interest.

Ethical Consideration: Informed consent was obtained from all the participants of the study.

Funding: No funding was received for this research.

Article History: Received: 01 February 2024. Revised: 23 April 2024. Accepted: 24 April 2024. First published: 25 April 2024

Published by: Aesthetix Media Services, India ↗

Citation: Tripathi, V & Rathod, D. (2024). Semantic Model for Fragment of Hindi (Part 2). *Rupkatha Journal* 16:2. https://doi.org/10.21659/rupkatha.v16n2.02

Introduction

Formal semantics is crucial in linguistic theory, providing a principled framework for analyzing language meaning. It is essential in studying natural language understanding, machine translation, and computational linguistics, where precise and unambiguous representations of meaning are crucial for developing algorithms and systems that can work with natural language. It helps bridge the gap between the richness and subtlety of human language and the need for clarity and precision in linguistic analysis.

Montague promoted the use of truth-conditional semantics to indicate the truth and falsity of a sentence as a parameter of the meaning of any well-formed formula (wff). Suppose a sentence is not a well-formed formula (wff); in that case, the computation system rejects the sentence based on syntactical rules, thus preventing the formation of non-grammatical sentences. However, suppose the sentence is not syntactically ambiguous. In that case, the semantic model allows us to determine every syntactical node's semantic value (SV), thus determining logical entailments and inferences based on said proposition.

This paper ventures into the intriguing domain of constructing a formal model for Hindi-like Fragment - a proposition that seeks to adapt and extend the principles of Montague semantics to accommodate the linguistic intricacies of Hindi and similar languages. The beauty of such an endeavour lies in the potential to unravel the complexities of these languages with the same mathematical precision that Montague semantics brought to English. Here, we have adopted the approach of Montague semantics, for which the first step is the rules of generative grammar.

We must first delve into the foundations of Montague semantics and its adaptability to non-English languages to embark on this linguistic journey. Then, we will explore the unique linguistic characteristics of Hindi and similar languages, highlighting the challenges that necessitate a specialized formal model. Various linguistic theories have been applied to Hindi, enriching our knowledge of its grammatical structure and linguistic characteristics (Kachru, 2008), encompassing its intricate case system, verb morphology, and diverse syntactic structures.

Formal models for Hindi-like fragments aim to capture these nuances while ensuring logical consistency and semantic precision. A syntactic parser and semantic analyzer that uses the rules of generative grammar and is compatible with the principle of compositionality will be an essential tool for our semantic analysis of Hindi in later sections. Works of Chomsky (Chomsky, 2009), Dowty (Dowty, 1982) and Barbara Partee (Partee, 1973) have laid the basic foundation of this research work.

In doing so, we aim to contribute to the broader discourse surrounding linguistic formalisms and provide a framework that facilitates a deeper understanding of Hindi and languages with similar complexities. Developing formal models for Hindi-like fragments holds significant promise for various applications. These applications extend to natural language processing (NLP), machine translation (MT), information retrieval (IR), and natural language understanding (NLU). Precise semantic representations enable machines to interpret better and generate human language, enhancing communication between humans and AI systems.

Semantic Analysis

A traditional notion within the field of semantics posits that its primary concern resides in elucidating word meanings. Under this perspective, the creation of dictionaries (a concept sometimes referred to as "semantics as lexicography") construes as the core function of semantics. In the contemporary computational understanding of any language, a variety of natural language processing (NLP) tasks use semantical ideas, such as machine translation, question answering, and sentiment analysis through different semantic models. One common type of semantic model is word embedding. A word embedding is a vector representation of a word that captures its meaning and relationships to other words. A large corpus of text helps machines learn word embeddings and perform various semantic tasks, such as word similarity, analogy detection, and sentiment analysis. Another type of semantic model is the semantic role labelling (SRL) model. An SRL model identifies the semantic roles of the words in a sentence. For example, in the sentence "*Billī caṭāī pe sotī hai*" (i.e. The cat sleeps on the mat), the SRL model would identify the *Billī* as the agent, the *caṭāī* as the patient and *sotī hai* as the verb.

However, this work has researched the topic of knowledge representation (KR) with Hindi as the target language, intending to develop AI semantics capabilities through logical inference rules explicitly applied to the Hindi language. A CFG proposed in the previous part (i.e. Part 1 of this research; check Tripathi, 2024) directly analyzes the syntax. In contrast, the current section (i.e. Part 2) will provide a model for the semantics of a fragment created through the formal characterization of finite lexical units of Hindi (a type of formal semantics called Montague semantics).

Among the recent research in Montague semantics, Roy et al., 2004 focused on extending Montague semantics to accommodate complex language structures, particularly n-ary transitive verbs. They used Montague semantics to construct natural-language processors in higher-order functional languages. Dupont et al., 1990 compare Montague's semantics with Boolean semantics for natural language representation with an attempt to apply some techniques of mathematical logic and algebraic lattice theory, respectively, for the representation of natural language with a view to its automatic processing. Warren et al., 1982 explore the use of semantics in non-context-free parsing of Montague grammar, explicitly addressing the reduction of syntactic ambiguity, while Purdy, 1990 discusses the need for lexical extension of Montague semantics to enhance the representation of meaning at the lexical level (as Montague theory does not deal with meaning at the lexical level due to which deduction in a system based on Montague semantics is severely restricted).

However, introducing formal logic to natural language gave us an understanding of meaning as equivalent to derived senses. These senses may denote any ideas (individual or predicate) of the natural world or any imaginary functional world. These computational worlds are usually denoted as 'possible worlds'. Only some ideas (like individuals) can accurately reference the real natural world. In contrast, few ideas (like predicates) can have no references. Thus, whatever sense a lexical unit represents is termed as its denotation.

These denotations are references for individuals and set-theoretic ideas for predicates. Therefore, in the formal study of language (like in this research), "semantics as denotation" is studied as its meaning. These denotations are pre-defined in the created possible world through a model.

Consequently, we are more inclined to derive the meaning of individual lexical units as denotations within the 'possible world' established by a computational system. The process of functional application (FA) provides for the interpretation of phrases and complete sentences, whereby the syntactic relations of words contribute to the semantic relationship between them rather than the reverse.

The semantic interrelationship among lexical units and the method of executing FA adhere to the 'principle of compositionality'. In formal linguistics, these formal semantic methods are mathematical and logical methods for representing and reasoning about the meaning of language, thus contributing to developing computational models of semantics. The formal cum computational methods can be used to analyze and reason about the meaning of natural language sentences.

The following section details the semantic rules and process of evaluating truth conditions and truth values of lexical tokens (of Hindi) based on the principle of compositionality. Many researchers have used Montague semantics (as listed in the upper part of the current section) to develop computational models of semantics for English and other languages. However, it has yet to be understood and its applicability to the natural language, Hindi. Montague semantics addresses a diverse range of semantic phenomena, encompassing reference, predication, quantification, modality, tense, and aspect. In the present research context, our L_{0H} fragment is primarily concerned with exploring reference and predication as its principal objectives.

We will adopt the characteristic function of S [Equation 1] to determine denotations of syntactic nodes, i.e., semantic values for each phase structure rule $\alpha \rightarrow \beta_1\ \beta_2\ \beta_3\ \beta_4 \ldots\beta_n$ proposed as per CFG in Part 1 of this research work (see Tripathi et al., 2024). In Montague semantics, a characteristic function, also called a denotation, is a mathematical function that assigns a semantic value to an expression. These functions are fundamental in the framework for representing the meaning of linguistic expressions precisely and formally.

Characteristic functions typically assign meanings to words, phrases, and sentences. For example, a characteristic function for a common noun might assign a set of individuals as its meaning. In contrast, for the meaning of any transitive verb, a characteristic function might be a function that takes two arguments and returns true (1) or false (0) based on whether the verb holds true for those two arguments. The semantic values of all syntactic nodes of a sentence will be combined based on the principle of compositionality, thus providing the semanticity of the whole string.

Suppose D is a set of individuals in the given world. We define S as a subset of D for our possible world to perform logical computation. In that case, we define a function f_s on subset S to identify the mapping of each *individual a* to 0, ie. false-ness or 1, ie. truth-ness based on membership of *a* in the subset S (S is usually a predicate in any proposition). This function is termed a characteristic function (f_s or f or X_s as in Equation 1), which checks on membership of "a" through 0 or 1 in subset S.

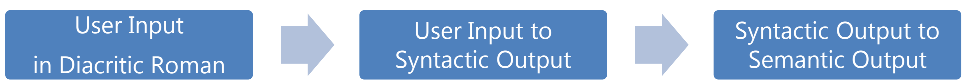
$$f_s(a) = f(a) = \begin{cases} 1 \text{ if } a \in S \\ 0 \text{ if } a \notin S \end{cases} \qquad \text{[Equation 1]}$$

When we lay out rules for object language L_{0H}, which is based on explicit syntax and semantics of predicate logic (of referential NPs), we use the indirect interpretation that Montague adopted (1973a) in his PTQ paper (Dowty et al.,1981 and Dowty et al.,1982) as shown in Table 1.

User Input (Hindi)	**Syntactic Output**	**Denotation i.e. Semantic Output**
rām	ram	I (ram) = ⟦ram⟧ = r
sītā	sita	I(sita) = ⟦sita⟧ = s
viveka	vivek	I(vivek) = ⟦vivek⟧ = v

Table 1. Indirect interpretation of Individuals

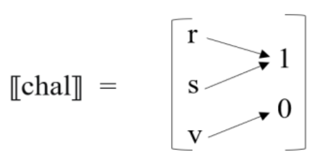
In Montague semantics, an interpretation function, often denoted as "I," is a crucial component of the framework used to formally represent the meaning of linguistic expressions. The interpretation function maps expressions from the language of natural discourse (e.g., English, Hindi, Japanese, etc.) to elements in a formal language. Its primary purpose is to provide a precise, mathematical description of the relationship between expressions in a natural language and their semantic interpretations in a formal language. This interpretation function (I) applies to syntactic outputs containing no diacritic marks (and not to Hindi input in our current research design), in the following ways (see flowchart 1):



Flowchart 1. Conversion of Hindi User Inputs to Semantic Outputs

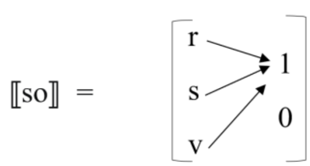
The interpretation function (I) serves to assign denotations (with denotation bracket ⟦ . ⟧) or semantic values to expressions. For example, it might assign set-theoretic entities, functions, or other mathematical objects as the meanings of words, phrases, and sentences in the natural language.

The semantic value of V_i in L_{0H} will be characteristic functions of sets of individuals. For example, the following two intransitive verbs can be defined in the following way:



[Set 1]

This function is technically a set of ordered pairs: thus, is simply a convenient graphic representation of the set {<r, 1>, <s, 1>, <v, 0>}.

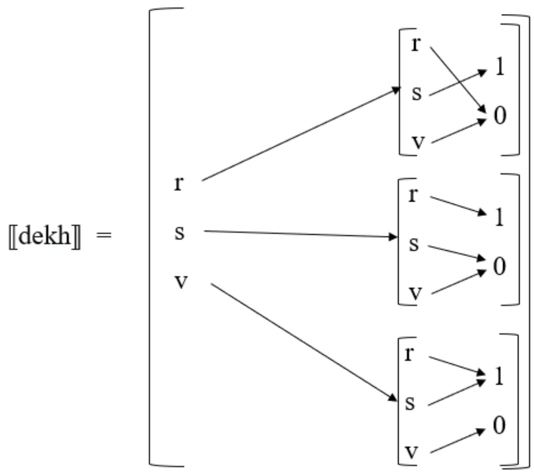


[Set 2]

This function is technically a set of ordered pairs: thus, is simply a convenient graphic representation of the set {<r, 1>, <s, 1>, <v, 1>}.

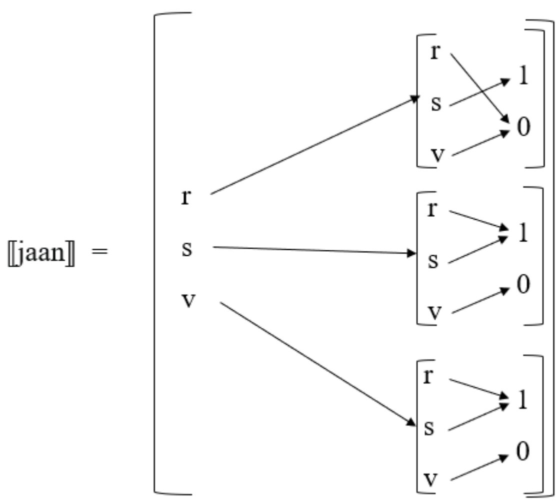
Semantically, V_t in L_{0H} seem to express relationship between individuals. According to grammar, V_t (as part of $Verb_t$ item) is preceded by N (of NP_o item) and followed by Aux. Thus, the semantic value of V_t should be something which can take preceding N into account and thus the characteristic function like of Set 3 and Set 4 can be defined.

Thus, we take the SV of V_t to be something that maps the semantic value of N (i.e. an individual) into the semantic value of VP (i.e. a function from individuals to truth values). For example, the V_t "dekh" and "jaan" might have the following semantic value:



[Set 3]

This function is technically a set of ordered pairs within the ordered pairs: thus, is simply a convenient graphic representation of the set {<r, <r, 0>>, <r, <s, 1>>, <r, <v, 0>> <s, <r, 1>>, <s, <s, 0>>, <s, <v, 0>>, <v, <r, 1>>, <v, <s, 1>>, <v, <v, 0>>}.



[Set 4]

This function is technically a set of ordered pairs within the ordered pairs: thus, is simply a convenient graphic representation of the set {<r, <r, 0>>, <r, <s, 1>>, <r, <v, 0>> <s, <r, 1>>, <s, <s, 1>>, <s, <v, 0>>, <v, <r, 1>>, <v, <s, 1>>, <v, <v, 0>>}.

Since sets and characteristic functions are essentially two ways of looking at what amounts to the same things, hence predicates can be written in multiple ways for better visualization of ideas defined within a predicate. Above f_s can be written in terms of I or in denotation brackets as:

- $f_{chal}(x) = I(chal) = [\![chal]\!] = \{r, s\}$
- $f_{so}(x) = I(so) = [\![so]\!] = \{r, s\}$
- $f_{dekh}(x) = I(dekh) = [\![dekh]\!] = \{<s, r>, <r, s>, <v, r>, <v, s>\}$
- $f_{jaan}(x) = I(jaan) = [\![jaan]\!] = \{<r, s>, <r, s>, <s, s>, <r, v>, <s, v>\}$

A definition like below, for denotation, can be stated more compactly:

$$[\![L]\!] = \text{the denotation of } L = f : \{x : x \text{ is an entity }\} \rightarrow \{1, 0\}$$

$$\text{thus, for every } y \in \{x : x \text{ is an entity }\},\ f(y) = 1 \text{ iff } y \text{ is a member of } [\![L]\!].$$

Following Montague's practice, we often present in terms of sets rather than functions when it is intuitively congenial. The semantic value of Aux in L_{0H} intuitively (explained in the upcoming section) is the characteristic function of the predicate itself. Thus, as per the phrase structure tree, it acts as a pointer to the predicate it associates itself with. So, Aux is calling back the predicate (precisely verb function) eg. $f_{chal}(x)$, $f_{so}(x)$, $f_{dekh}(x)$, $f_{jaan}(x)$ if Aux is associated with *chal, so, dekh, jaan* respectively; hence can be called as a callback function) and returns the same value (therefore, it can also be referred to as an identity function).

Further, it does not contain any semantic meaning (hence, it can be called an empty Function or semantically vacuous function). For example, the 'Aux' can be defined in the following way:

$$[\![Aux]\!] = \begin{bmatrix} [\![chal]\!] \longrightarrow [\![chal]\!] \\ [\![so]\!] \longrightarrow [\![so]\!] \\ [\![dekh]\!] \longrightarrow [\![dekh]\!] \\ [\![jaan]\!] \longrightarrow [\![jaan]\!] \end{bmatrix} \qquad \text{[Set 5]}$$

This function is technically a set of ordered pairs: thus, is simply a convenient graphic representation of the set $\{<[\![chal]\!], [\![chal]\!]>, <[\![so]\!], [\![so]\!]>, <[\![dekh]\!], [\![dekh]\!]>, <[\![jaan]\!], [\![jaan]\!]>\}$.

The semantic value of the Object in L_{0H} is also an identity function. It returns the same value as the individual it is attached to. Thus, as per the phrase structure tree, it acts as a pointer to the individual it associates with. So, Object is calling back the individual (regardless of the individual's gender), e.g. r, s, v if Object is associated with r, s and v, respectively. Hence, the Object can be referred to as a callback function) and returns the same value (hence, it can also be referred to as an identity function). Further, it does not contain any semantic meaning (hence, it can also be called an empty Function or semantically vacuous function).

For example, the 'Object' can be defined in the following way:

$$[\![\text{Object}]\!] = \begin{bmatrix} \mathbf{r} \longrightarrow \mathbf{r} \\ \mathbf{s} \longrightarrow \mathbf{s} \\ \mathbf{v} \longrightarrow \mathbf{v} \end{bmatrix} \qquad \text{[Set 6]}$$

This function is technically a set of ordered pairs: thus, is simply a convenient graphic representation of the set {<r, r>, <s, s>, <v, v>}.

This principle inherently maintains a close and intricate relationship with the syntactic structures of the language under consideration (e.g. Hindi here). The CFG proposed in Part 1 of this research provides us with two syntactic categories: eight lexical categories and six non-lexical categories. In the upcoming discussion, given the SV of both names and predicates, I will demonstrate the semantic computation for L_{0H} sentences.

Semantic Rules

Turning to the semantic rules of L_{0H}, we will provide a semantic rule for each syntactic rule used in producing sentences. To interpret the structure of the sentence "*ram chalta hai*" (see Figure 1), which will ultimately involve applying the function "*chalta hai*" to "*ram*", we need semantic rules for the phrase-structures rules that introduce the intervening nodes NP, VP, $Verb_i$, V_i and Aux.

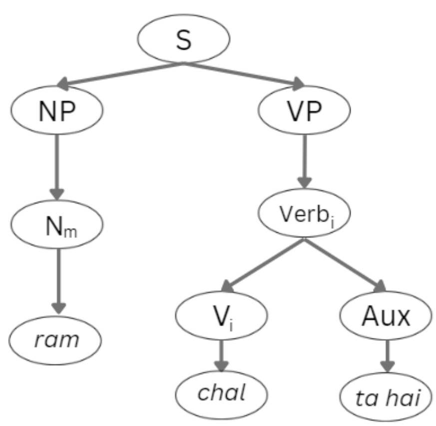


Figure 1. Phrase structure diagram for Intransitive Verb

The semantic value of the nodes labelled with lexical categories N_m, V_i and Aux should be the semantic values of the respective lexical items that they immediately dominate. Thus, the semantic role corresponding to the syntactic rule $N_m \rightarrow$ ram should be something like the following:

- If α is N_m and β is ram, then ⟦α⟧ = ⟦β⟧.

  |
  ram

The semantic rules corresponding to $V_i \to$ chal, Aux $\to$ ta hai would be similar. And in fact, it can abbreviate all such semantics by the means of the following rule schema:

Semantic Rule 1:

If α is γ, where γ is any lexical category and β is any lexical item and γ → β is a syntactic rule,
|
β

then ⟦α⟧ = ⟦β⟧.

For the grammar of L_{0H}, this semantic rule is instantiated by eight semantic rules, each corresponding to a lexical rule of the grammar. Corresponding to the non-lexical syntactic rules $Verb_i \to V_i$ Aux, a semantic rule attaches the semantic value of the V_i node to the $Verb_i$ node is required. (as Aux is a semantically vacuous function). In a way familiar to linguists, we use a triangle like $Verb_i$ to denote any tree rooted in $Verb_i$.

Semantic Rule 2:

If α is $Verb_i$ and β is semantically non-vacuous and γ is a semantically vacuous function,
/ \
β γ
then ⟦α⟧ = ⟦β⟧.

Semantic Rule 3:

If α is ϕ, where ϕ is a binary node (of β and γ) and β is semantically non-vacuous predicate
/ \
β γ
and γ is a semantically vacuous identity function, then ⟦α⟧ = ⟦β⟧.

Corresponding to the non-lexical syntactic rules VP $\to Verb_i$, a semantic rule which attaches the semantic value of the $Verb_i$ node to the VP node is required.

- If α is VP and β is $Verb_i$, then ⟦α⟧ = ⟦β⟧.
  |
  β

The semantic rules corresponding to other non-lexical syntactic rules, such as NP $\rightarrow N_m$, NP $\rightarrow N_f$, and VP $\rightarrow Verb_i$ would be similar. And in fact, it can abbreviate all such semantics by means of the following rule schema:

Semantic Rule 4:

If α is γ, and γ, β are any non-lexical items and γ → β is a syntactic rule, then ⟦α⟧ = ⟦β⟧.
|
β

Finally, we come to the most interesting semantic rule, which corresponds to the branching syntactic rule S → NP VP:

Semantic Rule 5:

If α is NP and β is VP, and if γ is S is then ⟦γ⟧ = ⟦β⟧(⟦α⟧).
/ \
α β

Thus, by using the schema of semantic rules 1 to 5, the semantic value of phrase structure tree of type "*ram chalta hai*" will check membership of r in set {r, s} and hence the rules determine the semantic value of given wff as 1 by using characteristics function $f_{chal}(r)$.

We now state the semantic rule corresponding to the syntactic rule VP → (NP_o) $Verb_t$ (and the structures it dominates) to illustrate the semantic computation for the phrase structure of the transitive type "sita ko dekhta hai" (see Figure 2). In the given case, NP_o and $Verb_t$ are binary branching nodes, and none are semantically vacuous.

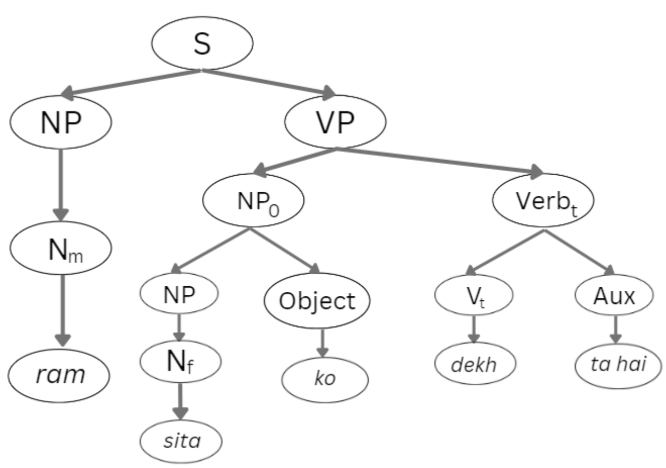


Figure 2. Phrase structure diagram for Transitive Verb

Corresponding to the syntactic rules $NP_o \rightarrow$ NP Object, the following semantic rule computes the semantic value of NP and Object:

Semantic Rule 6:

> If α is ϕ, where ϕ is a binary node (of β and γ) and β is an individual and γ is a semantically
> $\beta \quad \gamma$
> vacuous identity function, then $[\![\alpha]\!] = [\![\beta]\!]$.

Corresponding to the non-lexical syntactic rules $Verb_t \rightarrow V_t$ Aux, a semantic rule which attaches the semantic value of the V_t node to the $Verb_t$ node is applied using rule semantic rule schema 2. Thus, a semantic rule corresponding to the syntactic rule $VP \rightarrow (NP_o)\ Verb_t$, which can attach the semantic value of $Verb_t$ to VP, is required.

Semantic Rule 7:

> If α is NP_o and β is $Verb_t$, and γ is VP, then $[\![\gamma]\!]$ is $[\![\beta]\!]([\![\alpha]\!])$.
> $\alpha \quad \beta$

By this semantic rule, semantic value of "sita ko dekhta hai" can be determined as using SV of "dekhta hai" (i.e. {<r, <r, 0>>, <r, <s, 1>>, <r, <v, 0>> <s, <r, 1>>, <s, <s, 0>>, <s, <v, 0>>, <v, <r, 1>>, <v, <s, 1>>, <v, <v, 0>>}) applied to SV "sita ko" (i.e. s) as an argument resulting into following intermediate set as {<r,1>}. The phrase "sita ko dekhta hai" can also be written as:

$$[\![\text{sita ko dekhta hai}]\!] = \begin{bmatrix} r \rightarrow 1 \\ s \rightarrow 0 \\ v \rightarrow 0 \end{bmatrix} \qquad \text{[Set 7]}$$

This function is technically a set of ordered pairs: thus, is simply a convenient graphic representation of the set {<r, 1>, <s, 0>, <v, 0>}.

The computational algorithm generates this function as an intermediary function (a function from individual to truth value). Now, we are in a position to determine the semantic value of "*ram sita ko dekhta hai*" from its tree structure by applying the semantic rule schema 4, according to my assumption about $[\![$ram$]\!]$, $[\![$sita$]\!]$, $[\![$vivek$]\!]$ and the semantic rules. Thus, using schema 4., the semantic value of the phrase structure tree of type "ram sita ko dekhta hai" will check the membership of r in the above intermediate set {r} and hence, the semantic value is calculated as 1.

The remaining lexical items are the negation operators "aisa nahi hai ki" under two other conjunction operators "aur" and "ya". We will assume that these have semantic values corresponding to the logical connectives as follows (also see Table 2 for SVs):

- ¬ is equivalent to logical output of 'aisā nahī̃ hai ki'
- ∧ is equivalent to logical output of 'aur'
- ∨ is equivalent to logical output of 'yā'

A. aisa nahi hai ki

p	¬ p
1	0
0	1

B. aur

p	q	p∧q
1	1	1
1	0	0
0	1	0
0	0	0

C. ya

p	q	p∨q
1	1	1
1	0	1
0	1	1
0	0	0

Table 2. Semantic value of L_{0H} expressions in equivalence to logical constant

Syntactically, "aisa nahi hai ki" is combined with a sentence to form another sentence. Therefore, in L_{0H}, we may treat this phrase as a function mapping a truth value into a truth value, which is what truth table 4 is.

Semantic Rule 8:

If α is Neg and β is S, and if γ is S, then ⟦γ⟧ is ⟦α⟧(⟦β⟧).

α β

The logical constants ∧ and ∨ are actually two place connectives that take two inputs from a pair of propositions to yield one output. Thus, only one semantic rule is required, corresponding to the syntactic rule S → S Conj S.

Semantic Rule 9:

If α is Conj and β is S, and γ is S, and if ω is S, then ⟦ω⟧ is ⟦α⟧(<⟦β⟧, ⟦γ⟧>).

β α γ

Given all these semantic rules (1 to 9), now it is possible to assume semantic values for any terminal and non-terminal symbols of wffs of L_{0H} and, in particular, any sentence (of any length) of L_{0H}.

Application of Semantic Model: A Case Study

Through this case study, we aim to showcase the efficacy and versatility of our semantic model in elucidating the syntactic and semantic structures inherent in complex linguistic expressions of fragment L_{0H}. We will show the application of our model through a case study centred around the sentence: "ram sota hai aur vivek ram ko dekhta hai." (whose diacritical Roman input was "rāma sotā hai aura viveka rāma ko dekhtā hai.") The CFG rules of L_{0H} fragment generated the following parse tree (see Figure 3). This parse tree demonstrates the syntactic categories and provides nodes for applying semantic rules.

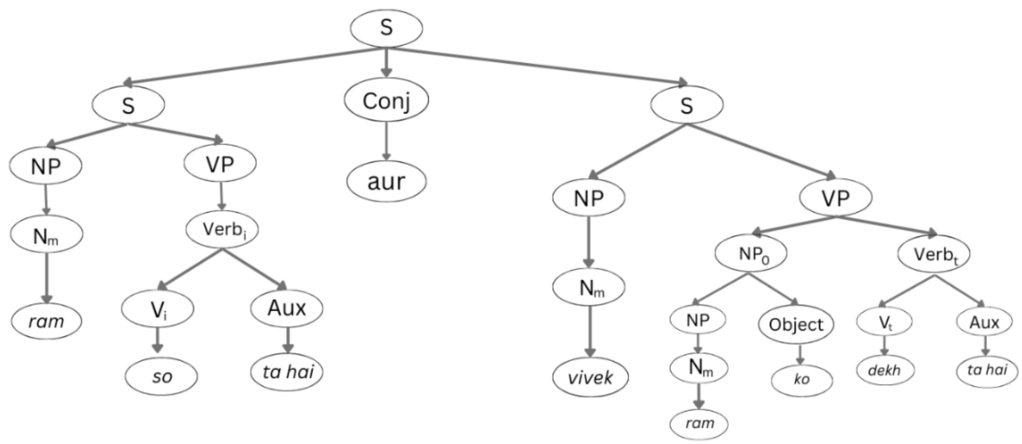


Figure 3. Phrase structure diagram for a complex sentence of L_{0H}

The aforementioned parse tree satisfies the criteria for being a well-formed formula (wff) of L_{0H}. Consequently, the syntactic configurations of nodes within the aforementioned parse tree are deemed suitable for evaluation according to the prescribed semantic rules 1 to 9. The application of these semantic rules is elucidated through the subsequent table presentation (see Table 3).

Column 1	Column 2	Column 3	Column 4
SV of Subtree S (LHS)	SV of Subtree S (RHS)	SV of Conj	SV of Tree S
Using semantic rules 1,2,3,4 and 5, and set 2 and 5 and Table 1 as a glossary, the semantic value of ‘ram sota hai’ is 1.	Using semantic rules 1,3,5,6 and 7, and set 3, 5 and 6 and Table 1 as a glossary, the semantic value of ‘vivek ram ko dekhta hai’ is 0.	Using Table 2 as the semantic value of ‘aur’.	Based on the SVs of the LHS subtree as 1, the RHS subtree as 0, and the SV of conjunction as $<<1,0>\rightarrow 0>$, the SV of Tree S is calculated as 0.

Table 3. Semantic Calculation Procedure from Column 1 to Column 4

Therefore, through the exemplification of an intransitive-based sentence (e.g., "ram chalta hai") alongside a transitive verb-based statement (e.g., "ram sita ko dekhta hai") and by delineating semantic rules about logical constants, our computational algorithm is proficient in formally computing any well-formed formula (wff) within the fragment L_{0H}. The selection of these two examples encapsulates all such propositions that the L_{0H} fragment can generate and semantically evaluate within its defined parameters.

Conclusion

After conducting a series of experiments outlined in this paper, we have reached several significant conclusions regarding our proposed formalization of Hindi with a basic fragment called L_{0H}. This fragment allows for syntactic analysis using context-free grammar (through Part 1) and semantic analysis using a model-theoretic system (through Part 2). In summary, the essential findings and insights from our experiments are as follows:

- Semantic Analysis of simple and complex propositions: Our model characterizes meaning in terms of individuals and sets, facilitating the syntactic distribution of well-formed formulas. The model allows for a robust computation cum analysis of syntactic nodes in terms of function and ultimately, computing the semantic value of propositions.

- Versatility: Our system is not limited to specific linguistic models; it can work with any model cum possible world described in a glossary. This adaptability makes it a valuable tool for linguistic analysis across various domains.

- Inter-Relations of Syntactic Category: The ability of our model to handle inter-relations further underscores its linguistic versatility for linguistic studies. It can capture the intricacies of meaning and relationships within the formal set up.

In conclusion, our rule-based syntactic and semantic model for the fragment of the Hindi language has demonstrated its effectiveness in handling various linguistic phenomena. The successful completion of these experiments reinforces the correctness and robustness of our formal model, making it a valuable contribution to the field of natural language processing, specifically for the Hindi language. We hope this research will serve as a foundation for further exploration and development in computational linguistics for Hindi and other languages with similar complexities

.

Note: On behalf of all authors, the corresponding author states that there is no conflict of interest.

Acknowledgement

On behalf of all contributors to this paper, I (Vivek Tripathi) would like to extend my sincerest gratitude to Dr. Nirmalya Guha for his invaluable feedback and guidance throughout developing this paper. The best aspects of this paper owe much to his direction, while any remaining mistakes are solely my own. I further thank Dr. Sanjukta Ghosh and Dr. V. Ramanathan for their constructive criticism and insightful suggestions that have significantly improved work quality. (All named persons are faculty at the Indian Institute of Technology, BHU, Varanasi).

References

Chomsky, N. (2009). Syntactic structures. In Syntactic Structures. De Gruyter Mouton.

Dupont, P., & Thayse, A. (1990). Montague's Semantics and Boolean Semantics for Natural Language Representation. Philips Journal of Research, 45(1), 41.

Dowty, D. R., Wall, R. E., Peters, S., Dowty, D. R., Wall, R. E., & Peters, S. (1981). The Grammar of PTQ. Introduction to Montague Semantics, 179-251.

Dowty, D. (1982). Grammatical relations and Montague grammar. In The nature of syntactic representation (pp. 79-130). Springer.

Montague, R. (1973). The proper treatment of quantification in ordinary English. In K.J.J. Hintikka, J.M.E. Moravcsik & P. Suppes (Eds.), *Approaches to natural language* (Synthese Library 49, pp. 221-242). Reidel.

Kachru, Y. (2008). "Hindi." In "The World's Major Languages" (2nd ed., pp. 565-582). Routledge.

Partee, B. (1973). Some transformational extensions of Montague grammar. Journal of Philosophical Logic, 2(4), 509-534.

Purdy, W. C. (1990). A lexical extension of Montague semantics.

Roy, M., & Frost, R. (2004). Extending montague semantics for use in natural-language database-query processing. In Advances in Artificial Intelligence: 17th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2004, London, Ontario, Canada, May 17-19, 2004. Proceedings 17 (pp. 567-568). Springer Berlin Heidelberg.

Tripathi, V. & Rathod, D. (2024). Semantic Model for Fragment of Hindi (Part 1). *Rupkatha Journal* 16:1. https://doi.org/10.21659/rupkatha.v16n1.03g

Warren, D. S., & Friedman, J. (1982). Using semantics in non-context-free parsing of montague grammar. American Journal of Computational Linguistics, 8(3-4), 123-138.

Vivek Tripathi is a linguistics scholar affiliated with the Indian Institute of Technology (BHU) Varanasi. The primary focus of his research is to investigate the syntax and semantics of South Asian Languages. He is currently dedicated to understanding Hindi as a contemporary language and Sanskrit as a classical language from linguistic perspectives and frameworks. The areas of his research are syntax, semantics, modality and negative polarity items (NPI).

Dinesh Rathod is a researcher at the Computer Science and Automation Department of the Indian Institute of Science, Bangalore. He is dedicated to researching computer graphics, algorithmic problem-solving and software development. Dinesh has published papers in computer science and related domains such as geomechanics.