A Formal Grammar of Hindi

Vivek Tripathi

26.02.2026

Supervisor: Nirmalya Guha

Indian Institute of Technology (BHU) Varanasi

Formal Semantics

- Corresponding to every syntactic rule there is a semantic rule.
- One can describe a small model by using a formal/ logical language, and it will have a linguistic counterpart that looks like a fragment of a natural language.
- That small formal model may be updated by adding rules, variables, individuals and functions. That updated model too will have a linguistic counterpart.
- This process is ideally open-ended.

Outline

Objectives of the Study

This thesis:

1. develops **a formal grammatical framework for Hindi** that systematically incorporates syntactic rules and semantic interpretations.
2. utilizes **set-theoretic concepts** for analyzing semantics of Hindi sentences.
3. illustrates the application of **logic in Hindi linguistic structures** by studying one-place predicates (e.g., intransitive verbs), two-place predicates (e.g., transitive verbs) and three-place predicates (e.g., ditransitive verbs).
4. provides **a computationally tractable method** for analyzing Hindi sentences using model-theoretic approaches.

Selected Literature

- Montague Grammar [Dowty et al. (2012)]: Introduced a formal system for natural language semantics, integrating logic and syntax.
- Formal Semantics [Heim and Kratzer (1998)]: Provides insights into function-based meaning computation in natural languages.
- Lambda Calculus in Linguistics [Carpenter (1998)]: Offers a foundation for treating linguistic expressions as mathematical functions. Although I have not documented the Lambda-based treatment of Hindi's categories, this work has contributed to a deeper understanding of functions.
- Elements of Formal Semantics [Winter (2016)]: Offers most useful introduction to formal semantics of natural language (specifically type theory).
- Hindi Grammar [Kachru (2006)]: Discusses the grammatical terminologies for studying structural complexities of Hindi, including verb agreement and case marking.
- Computational Linguistics Approaches [Jurafsky et al. (2020)]: Present computational methods for syntactic and semantic analysis in NLP.

REFERENCE

Overview of L_{1H}

- This chapter introduces a grammatical framework.
- The syntactic rules parse Hindi expressions and semantic rules gives meaning to it.
- The system uses concepts from set theory and logical operators to interpret the denotations of various linguistic expressions, such as words (like **rāma**) and phrases (like **sītā ko**, **jānatā hai**), which in turn facilitate the computation of the semantic values of complete sentences (like **rāma sītā ko jānatā hai**).
- Hindi expressions are evaluated for well-formedness rooted on the principles of propositional logic. I illustrate the grammar by analyzing one-place predicates (e.g., **cala**) and two-place predicates (e.g., **dekha**) with gender-specific individuals, such as **rāma** (masculine noun) and **sītā** (feminine noun).

Basic Foundations

- Developing Model
- Developing Vocabulary
- Characteristic Functions
- Denotations
- Lexical Categories and Non-Lexical Categories

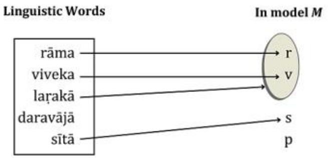


Figure 2.1: Model M maps Hindi linguistic words to abstract mathematical objects (AMO). *M* is a model with an entity denotation of ***rāma***, ***sītā***, a set denotation of ***laṛakā***. While ***daravājā*** has null AMO, the AMO *p* doesnot correspond to any syntactic category. The arrows designate the mappings from the Hindi linguistic words to their denotations, which are part of the model definition. The circle around AMOs *r* and *v* visually represents a set that includes *r* and *v*.

Syntax Rules

This grammar is capable of generating an infinite number of grammatical sentences by using the following recursive rules.

Syn 1: $S \rightarrow NP \quad VP$

Syn 2: $NP \rightarrow \begin{cases} N_m \\ N_f \end{cases}$

Syn 3: $VP \rightarrow \begin{cases} Verb_i \\ (NP_{cm}) \; Verb_t \end{cases}$

Syn 4: $Verb_i \rightarrow V_i \;\; Aux$

Syn 5: $NP_{cm} \rightarrow NP \; CM$

Syn 6: $Verb_t \rightarrow V_t \;\; Aux$

1. **Lexical Categories**: L_{1H} contains eight lexical categories - N_m, N_f, V_i, V_t, Aux, CM, Conj and Neg.

2. **Non-lexical Categories** (or Functional Categories): L_{1H} contains six non-lexical categories - S, NP, VP, $Verb_i$, $Verb_t$ and NP_{cm} stands for sentence, noun phrase, verb phrase, intransitive verb node, transitive verb node and case-marked noun phrase, respectively.

Syntax Rules

Syn 7: $Aux \rightarrow \begin{cases} \textbf{-tā hai} & \text{in the context } N_m \\ \textbf{-tī hai} & \text{in the context } N_f \end{cases}$

Syn 8: S $\rightarrow$ S Conj S

Syn 9: S $\rightarrow$ Neg S

Syn 10: S $\rightarrow$ **agara** S **to** S

Semantic Rules

Semantic Rule for Syntactic Rule 1 (S $\rightarrow$ NP VP):

Sem 1: Let α be an NP, β be a VP, γ is an S (sentence). The semantic value of γ is defined as: $\gamma = \beta\ (\alpha)$ i.e.,

$$[[\gamma]] \quad = \quad [[\beta]]\ ([[\alpha]])$$

where $[[VP]]$ is a function that takes $[[NP]]$ as its argument, indicating a function-argument relationship.

Semantic Rule for Syntactic Rule 2 (NP $\rightarrow$ N_m / N_f):

Sem 2: For any non-lexical categories γ and β, if $\gamma \rightarrow \beta$ is a syntactic rule, then:

$$[[\gamma]] = [[\beta]]$$

meaning that γ inherits the semantic value of β. For example, NP $\rightarrow$ N_m /N_f means the NP's denotation is that of the proper noun.

Semantic Rules

Semantic Rule for Syntactic Rule 3:

For Syntactic Rule 3a (VP $\rightarrow$ Verb$_i$):

Sem 3a: If VP $\rightarrow$ Verb$_i$, then:

$$[[VP]] = [[Verb_i]]$$

For Syntactic Rule 3b (VP $\rightarrow$ (NP$_{cm}$) Verb$_t$):

Sem 3b: Let α be NP$_{cm}$ (case-marked noun phrase), and β be Verb$_t$ (transitive verb node). If γ is VP, then:

$$[[\gamma]] = [[\beta]] \quad ([[\alpha]])$$

leads to [[VP]]=$[[Verb_t]]([[NP_{cm}]])$ meaning that the verb's semantic function applies to the case marked NP.

Semantic Rule for Syntactic Rule 4, 5 6 and 7 (Nodes with Identity Functions):

Sem 4: If $\alpha = \phi(\beta, \gamma)$ and γ is a semantically vacuous identity function, then:

$$[[\alpha]] = [[\beta]]$$

Here, γ does not contribute semantically, so α's meaning is inherited from β.

Semantic Rule for Terminal Nodes (containing Lexical Items):

Sem 5: For any terminal node γ with lexical item β, if $\gamma \rightarrow \beta$, then:

$$[[\gamma]] = [[\beta]]$$

Semantic Rules

Semantic Rule for Syntactic Rule 8 ($S \rightarrow$ S Conj S):

Sem 6: If S $\rightarrow$ S_1 Conj S_2, then:

$$[\![S]\!] \quad = \quad [[Conj]] \quad ([[S1]], [\![S2]\!])$$

where the conjunction function takes two sentence meanings as arguments and returns their conjunction.

Semantic Rule for Syntactic Rule 9 ($S \rightarrow$ Neg S):

Sem 7: If S $\rightarrow$ Neg S, then:

$$[\![S]\!] \quad = \quad [[Neg]] \quad ([[S]])$$

where [[*Neg*]] negates the meaning of the embedded sentence.

Semantic Rules

Semantic Rule for Syntactic Rule 10 (S $\rightarrow$ agara S to S)

Sem 8: If S $\rightarrow$ **agara** S_1 **to** S_2, then:

$$[\![S]\!] \quad = \quad [[Imp]] \quad ([\![S1]\!], [\![S2]\!])$$

where: $[[Imp]]$ is the semantic value of implication (i.e., the function that captures the truth-functional interpretation of **agara...to**). $[[S_1]]$ is the semantic value of the antecedent.[$[S_2]]$ is the semantic value of the consequent.

Overview of L_{2H}

- This chapter provides an in-depth analysis of noun phrases (NPs) in Hindi.
- In this context, I am going to introduce quantified NPs such as **hara laṛakā**, and quantifiers such as **hara** and **koī**.
- For various reasons, the value assignment function g of Montague Semantics cannot be used in our model directly. We are proposing an alternative method which is slightly different from Montague's.

Basic Foundation

- Understanding Predicate Logic
- Understanding Formulas and Quantifiers
- Value Assignment to Variables and the Gender Rules of Hindi

1. In the statement ***rāma calatā hai***, the expression ***cala*** (represented by $C(x)$) functions as a predicate that describes a property of ***rāma*** (represented by r). Formally, this can be represented as $C(r)$.

2. In the statement ***rāma sītā ko jānatā hai***, the expression ***jāna*** (represented by $J(x,y)$) functions as a predicate that describes a relation between Ram and Sita. Formally, this can be represented as $J(r,s)$.

1. **Universal Quantifier ($\forall$):** It means 'for all' or 'every'. It asserts that a predicate is *true* for all elements in the model. An expression ***hara*** corresponds to the universal quantifier in L_{2H}.

2. **Existential Quantifier ($\exists$):** It means 'there exists' or 'at least one'. It asserts that there is at least one element in the model for which the predicate is *true*. An expression ***koī*** corresponds to the existential quantifier in L_{2H}.

The syntax of L_{2H} is formalized through a phrase structure grammar, which consists of 12 syntactic rules. The ten syntactic rules (**Syn 1** - **Syn 10**) referenced here are those originally proposed in L_{1H}. In addition to the existing rules, I propose a total of three new rules: two entirely new rules and one modification of the **Syn 7** rule. These additions aim to establish a comprehensive grammatical framework for quantified expressions in Hindi.

The modified rule is a revision of Syn 7; thus, now it is as follows:

Syn 7:

$$Aux \rightarrow \begin{cases} \textbf{-tā hai} \text{ in the context } N_m \text{ or } CN_m \text{ outside the VP} \\ \textbf{-tī hai} \text{ in the context } N_f \text{ or } CN_f \text{ outside the VP} \end{cases}$$

Syntax Rules

The categories CN_m (common noun masculine) and CN_f (common noun feminine) are introduced as distinct syntactic categories, functioning as lexical categories in L_{2H}, analogous to N_m and N_f in L_{1H}. The CN_m and CN_f are derived from the CN (common noun) using the following rule, designated as **Syn 12**.

Syn 12: $CN \rightarrow \left[\begin{array}{l} CN_m \\ CN_f \end{array} \right.$

The syntactic category CN represents a common noun in the fragment L_{2H}. It is derived from the noun phrase (NP) itself through the application of the following rule, designated as Syn 11.

Syn 11: $NP \rightarrow \left[\begin{array}{ll} \textbf{hara} & CN \\ \textbf{koī} & CN \end{array} \right.$

Here, the category CN is quantified using syncategorematic elements such as **hara** and **koī.**

Semantic Rules

Sem 9: For any linguistic expression that is formed by combining a common noun (CN) modified by the quantifier **hara** and a VP,

$$[[\textbf{hara} \quad CN \quad VP]] = 1$$

iff for every entity e, if

$$[[CN]](e) = 1$$

then

$$[[VP]](e) = 1$$

.

Sem 10: For any linguistic expression that is formed by combining a common noun (CN) modified by the quantifier **koī** and a VP,

$$[[\textbf{koī} \quad CN \quad VP]] = 1$$

iff there exists at least one entity e, such that

$$[[CN]](e) = 1$$

and

$$[[VP]](e) = 1$$

Semantic Rules

Assigning values to variablesIn the Montagovian semantics, 'Every CN VP' and 'Some CN VP' are derived from their unquantified form, i.e., 'x VP'. Thus both 'Every boy knows Ram' and 'Every girl knows Ram' are derived from 'x knows Ram.'

- In English, the VP does not have gender agreement with the NP. But Hindi has gendered VPs (such as **jānatā hai** and **jānatī hai**).The Hindi counterpart of 'x knows Ram' is '**x rāma ko jānatā hai** / **jānatī hai**'.
- In Hindi semantics, value assignment to variables depends on the gender agreement rule. That is why our treatment of this is slightly different from the Montagovian treatment.

- Compare our Sem 9 with its counterpart in the Montagovian model. For instance, given any formula ϕ containing a single variable, such as x , and domain individuals a and b, the Montagovian semantics define the assignment function g that assigns individuals to variables and returns the truth value.

Semantic Rules

6. If α is a *CN*, u is a variable, and ϕ is a *For* containing at least one occurrence of u, then for ϕ' as in syntactic rule B6, $[\![\phi']\!]^{M,g} = 1$ iff for some value assignment g_u^e such that $[\![\alpha]\!]^{M,g}(e) = 1$, $[\![\phi]\!]^{M,g_u^e} = 1$.
7. If α is a *CN*, u is a variable, and ϕ is a *For* containing at least one occurrence of u, then for ϕ' as in syntactic rule B7, $[\![\phi']\!]^{M,g} = 1$ iff there is exactly one e in A such that $[\![\alpha]\!]^{M,g}(e) = 1$, and furthermore $[\![\phi]\!]^{M,g_u^e} = 1$.

Figure: Snapshot of Montagovian Semantic Rules 6 and 7

Overview of L_{3H}

- This chapter introduces basic mathematical concepts of types in the formal semantics of Hindi.

- I organize denotations into domains of different types (D_e and D_t). This general type system helps describe sets, relations, and operators with multiple arguments.

- Complex expressions are interpreted using a uniform function application (FA). I demonstrate this by analyzing one-place predicates (e.g., **cala**), two-place predicates (e.g., **dekha**), and three-place predicates (e.g., **de**).

- These predicates are utilized in both simple and complex sentence formation, contributing to a structured understanding of types of sentential connectives in Hindi. I further evaluated the type semantics of light verbs in Hindi and argued for its formalization using **cala** and **dekha**.

Basic Foundation

- Understanding Types and Domains
- Type Combination
- Type-theoretic representations for one-place two-place, and three-place predicates
- Issues associated with Light Verbs of Hindi

In this research, I adhere to the notation of writing denotations alongside their corresponding types for clarity and readability.

Analysing 1pp

In our analysis of the simple example **rāma calatā hai**:

$$[\![\textbf{tā hai (cala)}]\!]: \langle e, t\rangle + [\![\textbf{rāma}]\!]: \langle e\rangle = [\![\textbf{tā hai (cala)}(\textbf{rāma})]\!]: \langle t\rangle$$

$$[\![\textbf{cala}]\!]: \langle e, t\rangle + [\![\textbf{tā hai}]\!]: \langle\langle e, t\rangle, \langle e, t\rangle\rangle = [\![\textbf{tā hai (cala)}]\!]: \langle e, t\rangle$$

In our analysis of the simple example **rāma sītā ko dekhatā hai**:

⟦**tā hai (dekha) (ko (sītā))** ⟧: ⟨e, t⟩ + ⟦**rāma**⟧: ⟨e⟩ = ⟦**tā hai (dekha) (ko (sītā)) (rāma)**⟧: ⟨t⟩

⟦**tā hai (dekha)**⟧: ⟨e, ⟨e, t⟩⟩ + ⟦**(ko (sītā))**⟧: ⟨e⟩= ⟦**tā hai (dekha) (ko (sītā))** ⟧: ⟨e, t⟩

⟦**ko**⟧: ⟨e, e⟩ + ⟦**sītā**⟧: ⟨e⟩ = ⟦**ko (sītā)**⟧: ⟨e⟩

⟦**tā hai**⟧: ⟨⟨e, ⟨e, t⟩⟩, ⟨e, ⟨e, t⟩⟩⟩ + ⟦**dekha**⟧: ⟨e, ⟨e, t⟩⟩ = ⟦**tā hai (dekha)**⟧: ⟨e, ⟨e, t⟩⟩

Analysing 3pp

$[\![$**tā hai (de) (pustaka)(ko(sītā))**$]\!]$: ⟨e, t⟩ + $[\![$**rāma**$]\!]$: ⟨e⟩ = $[\![$**tā hai (de) (pustaka)(ko(sītā))(rāma)**$]\!]$: ⟨t⟩

$[\![$**tā hai (de) (pustaka)**$]\!]$: ⟨e, ⟨e, t⟩⟩ + $[\![$**(ko(sītā))**$]\!]$: ⟨e⟩= $[\![$**tā hai (de) (pustaka)(ko(sītā))**$]\!]$: ⟨e, t⟩

$[\![$**ko**$]\!]$: ⟨e, e⟩ + $[\![$**sītā**$]\!]$: ⟨e⟩ = $[\![$**ko(sītā)**$]\!]$: ⟨e⟩

$[\![$**tā hai (de)**$]\!]$: ⟨e, ⟨e, ⟨e, t⟩⟩⟩ + $[\![$**pustaka**$]\!]$: ⟨e⟩ = $[\![$**tā hai (de) (pustaka)**$]\!]$: ⟨e, ⟨e, t⟩⟩

$[\![$**tā hai**$]\!]$: ⟨⟨e, ⟨e, ⟨e, t⟩⟩⟩, ⟨e, ⟨e, ⟨e, t⟩⟩⟩⟩ + $[\![$**de**$]\!]$: ⟨e, ⟨e, ⟨e, t⟩⟩⟩ = $[\![$**tā hai (de)**$]\!]$: ⟨e, ⟨e, ⟨e, t⟩⟩⟩

Analyzing Negation

[[**aisā nahī̃ hai ki**]]: ⟨t, t⟩ + [[**tā hai (cala) (rāma)**]]: ⟨t⟩ = [[**aisā nahī̃ hai ki(tā hai (cala) (rāma))**]]: ⟨t⟩

[[**tā hai (cala)**]]: ⟨e, t⟩ + [[**rāma**]]: ⟨e⟩ = [[**tā hai (cala)(rāma)**]]: ⟨t⟩

[[**cala**]]: ⟨e, t⟩ + [[**tā hai**]]: ⟨⟨e, t⟩, ⟨e, t⟩⟩ = [[**tā hai (cala)**]]: ⟨e, t⟩

Analysing Sentential Connectives

$$[[\textbf{calatā hai (rāma)}]]: \langle t \rangle + [[\textbf{sotī hai (sītā)}]]: \langle t \rangle + [[\textbf{aura}]]: \langle t, \langle t, t \rangle \rangle = [[\textbf{aura (calatā hai (rāma)) (sotī hai (sītā))}]]: \langle t \rangle$$

$$[[\textbf{tī hai (so)}]]: \langle e, t \rangle + [[\textbf{sītā}]]: \langle e \rangle = [[\textbf{tī hai (so)(sītā)}]]: \langle t \rangle$$

$$[[\textbf{tā hai (cala)}]]: \langle e, t \rangle + [[\textbf{rāma}]]: \langle e \rangle = [[\textbf{tā hai (cala)(rāma)}]]: \langle t \rangle$$

$$[[\textbf{so}]]: \langle e, t \rangle + [[\textbf{tī hai}]]: \langle \langle e, t \rangle, \langle e, t \rangle \rangle = [[\textbf{tī hai (so)}]]: \langle e, t \rangle$$

$$[[\textbf{cala}]]: \langle e, t \rangle + [[\textbf{tā hai}]]: \langle \langle e, t \rangle, \langle e, t \rangle \rangle = [[\textbf{tā hai (cala)}]]: \langle e, t \rangle$$

Limitation of TT

- Butt (1995)'s monograph provided a systematic formal syntactic and semantic analysis of what she called 'light verbs'. In this monograph[1], she doesn't give a single dictionary-style definition of 'light verb' as a standalone term in one sentence. Instead, she theoretically develops the concept throughout the book by describing their function in complex predicates.
- For the purpose of this work, I am particularly interested in compounding and use a simple definition of light verbs as follows: Light verbs (LVs) are a class of verbs that combines with a main verb (MV), leading to a Compound Verb (CV) formation. Building on this idea, a Hindi infinitive can be defined as an infinitival form (marked by the suffix -nā) that consists of one main verb (MV), and followed by one or more light verbs (LVs), arranged sequentially from left to right. For instance:

$$[\mathrm{MV}\ ] + [\mathrm{LV1} + \mathrm{LV2} + \mathrm{LV3} + \ldots\ldots + \mathrm{LVn}] + \textbf{-nā}$$

[1] The book sets out criteria, properties, and cross-linguistic comparisons, solidifying the term 'light verb' in South Asian linguistic scholarship, for many researchers to devise their method of analysis on LVs.

Limitation of TT

- The syntactic analysis of an LV does not seem to have a straight forward semantic counterpart. Here is the reason. Let us compare: (a) **rāma calatā hai** (Rama walks), and (b) **rāma cala letā hai** (Rama can/is able to walk). Both **calatā hai** and **cala letā hai** here have the same semantic type: ⟨e, t⟩. The semantic type of **cala le** cannot be split further (semantically in the current setup). And surely, [[**calatā hai**]] and [[**cala letā hai**]] are different functions (or sets).

- This chapter on types shows us that sometimes syntactic analysis takes a certain path which semantics cannot follow. Such an unanalyzability limitation can be visualized in the figure on the next slide.

Limitation of TT

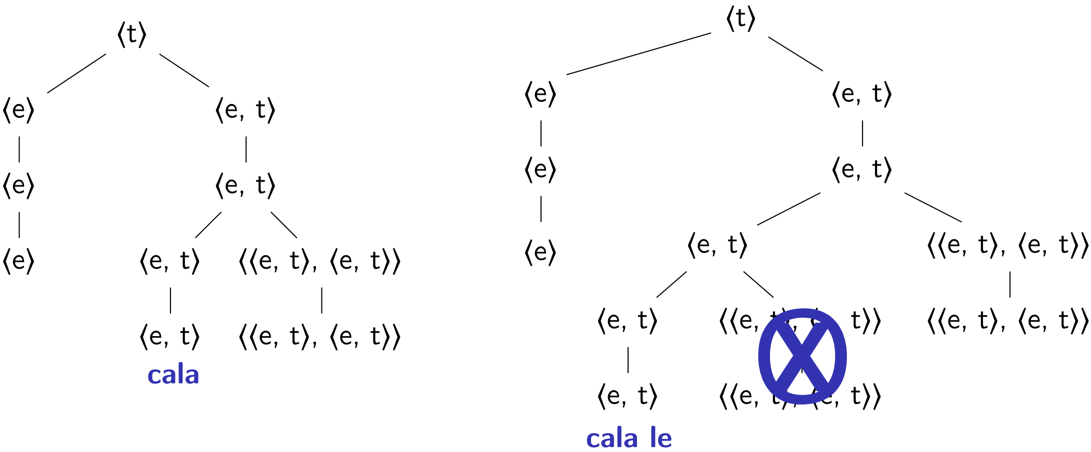


Figure: Semantic unanalyzability of **cala le** results in identical semantic types for **cala** and **cala le**.

Overview of L_{4H}

- This chapter introduces the concept of tense in Hindi propositions.
- It formalizes the notion of time using temporal logic (TL) and analyses tense markers and aspect markers in Hindi.
- The denotations of predicates are categorized into different models- M for past, present and future moments - allowing a structured analysis of Hindi tense markers for both intransitive predicates (like **cala**) and transitive predicates (like **dekha**).

Basic Foundation

- Understanding Time
- Understanding Temporal Logic

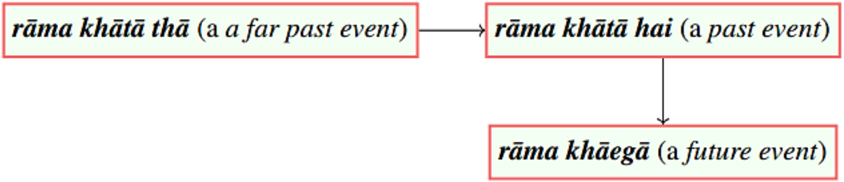


Figure 5.3: Linear sequencing of event based Hindi sentences

Syntax Rules

Syn 15: If ϕ is a Formula containing **tā hai** or **tī hai** (but not both), then ϕ' is a Formula where ϕ' comes from ϕ by replacing **tā hai** with **ā thā** and **tī hai** with **ī thī**.

Syn 16: If ϕ is a Formula containing **tā hai** or **tī hai** (but not both), then ϕ' is a Formula where ϕ' comes from ϕ by replacing **tā hai** with **egā** and **tī hai** with **egī**.

Sem 12: If ϕ is a Formula containing **tā hai** or **tī hai**, then for ϕ' as in Syn 15, $[[\phi']]^{M,t_p} = 1$ iff for at least one value of i, $[[\phi']]^{M,t_i} = 1$, when i $<$ p.

Sem 13: If ϕ is a Formula containing **tā hai** or **tī hai**, then for ϕ' as in Syn 16, $[[\phi']]^{M,t_p} = 1$ iff for at least one value of i, $[[\phi']]^{M,t_i} = 1$, when i $>$ p.

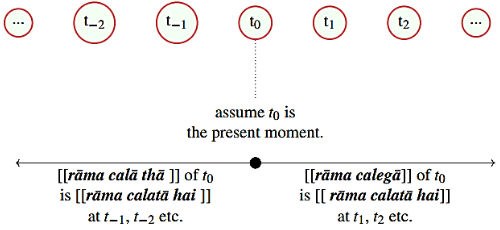


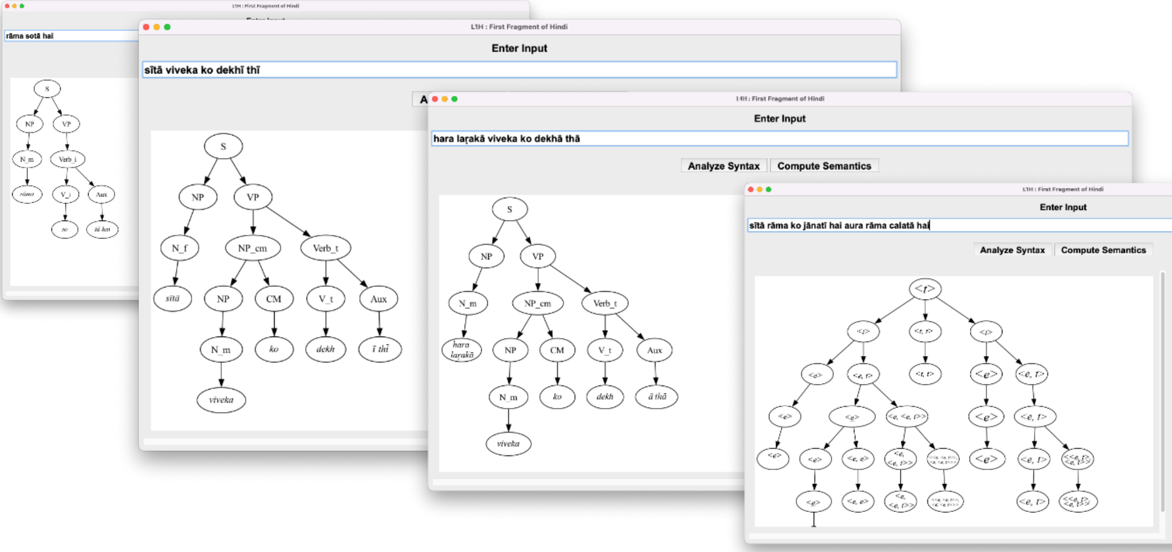
Figure: Present Tense is the core representation in Future and Past Tense based sentences.

- Hint Sentences - rāma calatā hai, rāma sītā ko jānatā hai, rāma sītā ko nahī̃ jānatā hai, rāma sītā ko dekhatā hai, aisā nahī̃ hai ki sītā calatī hai, rāma calatā hai yā śyama rāma ko sotā hai, rāma calatā hai yā śyāma sotā hai, sītā rāma ko dekhatī hai aur rāma calatā hai, śyāma rāma ko jānatā hai, aisā nahī̃ hai ki sītā sotī hai, aisā nahī hai ki sītā sotī hai aura rāma sītā ko dekhatā hai etc.

- This chapter presents a formal grammatical framework for the syntactic and semantic analysis of Hindi. The grammar developed through various components, namely L_{1H}, L_{2H}, L_{3H} and L_{4H} culminates in the final fragment, the 'Hindi Fragment', wherein all elements are integrated to form a comprehensive grammatical framework. This framework comprises 18 syntactic rules and 14 semantic rules, which facilitate the generation of valid, well-formed expressions in Hindi. In this chapter, my colleague[3] and I developed a software tool capable of performing syntactic parsing and semantic evaluation.

[3]Dinesh Rathod, IISc Bangalore

Software Demonstration



Conclusions

- This thesis presents a formal grammatical framework for the syntactic and semantic analysis of Hindi, grounded in concepts from formal Montague semantics.
- The framework interprets the denotations of various linguistic expressions,including individual words (e.g., **rāma**) and phrases (e.g., **sītā ko dekhatā hai**).
- The analysis encompasses one-place predicates (e.g., **cala**), two-place predicates (e.g., **dekha**) with a focus on gender agreement rules.
- Quantified noun phrases (NPs) are modelled as complex structures incorporating generalized quantifiers such as **hara** and **koī**.

Conclusions

- This thesis further outlines basic mathematical techniques, employing type theory to categorize one-place, two-place, and three-place predicates in Hindi. Denotations are systematically organized into domains of distinct types (e.g., D_e and D).
- Temporal logic (TL) is utilized to formalize the concept of time, enabling an analysis of tense and aspect markers.
- Additionally, a Python-based software tool is introduced, which implements the complete grammatical model. The grammatical rules are translated into an algorithmic design following a top-down approach for sentence processing. The software produces parsing trees and features a user interface (UI) for producing parsing trees and semantic computation, providing a practical demonstration of a developed theoretical framework.

Contributions

- L_{1H} – Propositional Logic-Based Fragment
- L_{2H} - Predicate Logic-Based Fragment
- L_{3H} - Type Theory-Based Fragment
- L_{4H} - Temporal Logic-Based Fragment

- THT Analyser System - The analyser, named The Hindi Tree (THT),automatically generates syntax trees that allows semantic computations for Hindi sentences. It parses annotated sentences and displays all possible trees that conform to the proposed grammatical rules for the final 'Hindi Fragment'. For open-source code of the tool, it will be available at iamalinguist.github.io

Scope

1. Extension to Complex Sentence Structures
2. Expansion to Multi-Verb Constructions
3. Cross-Linguistic Application
4. Machine Translation and NLP
5. Further Refinement of Temporal Logic

Publications

- Tripathi, V. (2024), 'Large Language Models: A technological approach for teaching Hindi', in G. Sharma and S. Bhatt (eds), *Advances in Hindi language teaching: Historical and methodological dimensions*, Chapter 2, pp. 69–114, LINCOM GmbH, Munich.

- Tripathi, V., and Rathod, D. (2024), 'Semantic model for fragment of Hindi (Part 1)', *Rupkatha Journal* 16(1).

- Tripathi, V., and Rathod, D. (2024), 'Semantic model for fragment of Hindi (Part 2)', *Rupkatha Journal* 16(2).

Deliverables of Research

1. Online Syntactic Parser
2. Offline Semantic Analyser

Accolades of Research

1. Best Research Award (by IIT Jodhpur & IIT Indore)
2. Top 10 Research Posters (in AI theme) before honorable PM, ESTIC 2025.

Thank you

for your time and support.

Bibliography (for Slide 5)

- Dowty, D. R., Wall, R. E., and Peters, S. (1981). *Introduction to Montague semantics*. Dordrecht: D. Reidel Publishing Company.
- Heim, I., and Kratzer, A. (1998). *Semantics in generative grammar*. Oxford: Blackwell.
- Carpenter, B. (1998). *Type-logical semantics*. Cambridge, MA: MIT Press.
- Winter, Y. (2016). *Elements of formal semantics: An introduction to the mathematical theory of meaning in natural language*. Edinburgh: Edinburgh University Press.
- Kachru, Y. (2006). *Hindi*. Amsterdam: John Benjamins Publishing Company.
- Jurafsky, D., and Martin, J. H. (2023). *Speech and language processing* (3rd ed.). Stanford University. Available at: https://web.stanford.edu/~jurafsky/slp3/

BACK