Analysis and Graphical Representation of Navya-Nyāya-Expressions
न्यायचित्रदीपिका

Arjuna S.R. and Amba Kulkarni

Department of Sanskrit Studies,
University of Hyderabad,
Hyderabad

World Sanskrit Conference-2015
Silpakorn University, Bangkok.

Supported by: Dr.Tirumala Kulkarni Acharya and his disciples

Navya-Nyāya

Navya-Nyāya (Neo-Logic)

- Branch of Indian school of Logic(Nyāyaśāstram).
- Udayana(10^{th} Century) initiated.
- Gaṅgeśa (12^{th} Century) gave a concrete form.

Navya-Nyāya

Navya-Nyāya (Neo-Logic)

- Branch of Indian school of Logic(Nyāyaśāstram).
- Udayana(10^{th} Century) initiated.
- Gaṅgeśa (12^{th} Century) gave a concrete form.

Contribution:

- Evolved a special language to deal with verbal cognition, logic and epistemology.
- Used by several important disciplines.

Navya-Nyāya

Navya-Nyāya (Neo-Logic)

- Branch of Indian school of Logic(Nyāyaśāstram).
- Udayana(10^{th} Century) initiated.
- Gaṅgeśa (12^{th} Century) gave a concrete form.

Contribution:

- Evolved a special language to deal with verbal cognition, logic and epistemology.
- Used by several important disciplines.

Special features of this language:

- Long compounds,
- Productive usage of secondary suffixes(Taddhita),
- Technical terminology.

Navya-Nyāya

Navya-Nyāya (Neo-Logic)

- Branch of Indian school of Logic(Nyāyaśāstram).
- Udayana(10^{th} Century) initiated.
- Gaṅgeśa (12^{th} Century) gave a concrete form.

Contribution:

- Evolved a special language to deal with verbal cognition, logic and epistemology.
- Used by several important disciplines.

Special features of this language:

- Long compounds,
- Productive usage of secondary suffixes(Taddhita),
- Technical terminology.

Problems:

- A Navya-Nyāya Expression(NNE) running into pages is hard to comprehend.

An NNE

Here is an example of an NNE involving a compound with ten components:

An NNE

Here is an example of an NNE involving a compound with ten components:
samavāyasambandhāvacchinnagandhatvāvacchinnagandha niṣṭhādheyatānirūpitādhikaraṇatāvatī.

समवायसम्बन्धावच्छिन्नगन्धत्वावच्छिन्नगन्धनिष्ठाधेयतानिरूपिताधिकरणतावती

Analysis of NNEs

The analysis of NNE involves following three steps.

Analysis of NNEs

The analysis of NNE involves following three steps.

1. Padaccheda.(Segmentation)

Analysis of NNEs

The analysis of NNE involves following three steps.

1. Padaccheda.(Segmentation)
2. Constituency Parsing of a compound.

Analysis of NNEs

The analysis of NNE involves following three steps.

1. Padaccheda.(Segmentation)
2. Constituency Parsing of a compound.
3. Transferring the phrase structure into a Conceptual Graph.

Cont...

For instance,

An NNE -

samavāyasambandhāvacchinnagandhatvāvacchinnagandha niṣṭhādheyatānirūpitādhikaraṇatāvatī.

Segmentation -

samavāyasambandha-avacchinna-gandhatva-avacchinna-gandha-niṣṭha-ādheyatā-nirūpita-adhikaraṇatā-vatī.

Constituency Parse -

<<<<<samavāyasambandha-avacchinna>-<<gandhatva-avacchinna>**T3**-<<gandha-niṣṭha>**T7**-ādheyatā>**K**>**K**>**K**-nirūpita>**T3**-adhikaraṇatā>**K**-∧vatī>[1]

[1] *K*, *T*3, *T*6, and *T*7 stand for *karmadhāraya*, *tatpuruṣa* with instrumental case, *tatpuruṣa* with genitive case and *tatpuruṣa* with locative case suffix respectively.

Cont...

Finally the Conceptual Graph -

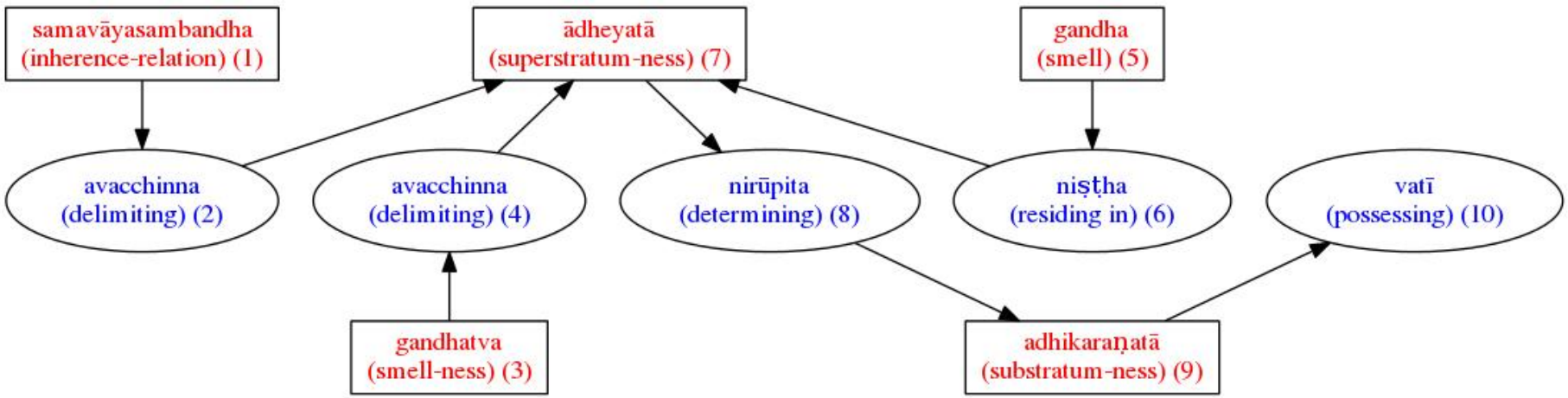


Figure : Graphical representation of the NNE

Segmentation

For the first step - Segmentation of NNE, Two important tools -

1. Saṃsādhanī(SCL)-splitter tool[2]

[2] http://sanskrit.uohyd.ernet.in/scl/sandhi_splitter/index.html
[3] http://sanskrit.inria.fr/DICO/reader.fr.html

Segmentation

For the first step - Segmentation of NNE, Two important tools -

1. Saṃsādhanī(SCL)-splitter tool[2]
2. Sanskrit Heritage Segmenter[3].

[2] http://sanskrit.uohyd.ernet.in/scl/sandhi_splitter/index.html
[3] http://sanskrit.inria.fr/DICO/reader.fr.html

Segmentation

For the first step - Segmentation of NNE, Two important tools -

1. Saṃsādhanī(SCL)-splitter tool[2]
2. Sanskrit Heritage Segmenter[3].

Initially, Sanskrit Heriatge Reader enhanced with recall of 91%.

[2] http://sanskrit.uohyd.ernet.in/scl/sandhi_splitter/index.html

[3] http://sanskrit.inria.fr/DICO/reader.fr.html

Segmentation

For the first step - Segmentation of NNE, Two important tools -

1. Saṃsādhanī(SCL)-splitter tool[2]
2. Sanskrit Heritage Segmenter[3].

Initially, Sanskrit Heriatge Reader enhanced with recall of 91%.
Later, SCL-splitter extended with a recursive greedy approach to get same recall as Heritage.

[2] http://sanskrit.uohyd.ernet.in/scl/sandhi_splitter/index.html
[3] http://sanskrit.inria.fr/DICO/reader.fr.html

SCL-NN Segmenter with controlled lexicon

Special morphological analyser with the vocabulary of Nyāya texts used in new version.

SCL-NN Segmenter with controlled lexicon

Special morphological analyser with the vocabulary of Nyāya texts used in new version.

Result:

100% recall.

SCL-NN Segmenter with controlled lexicon

Special morphological analyser with the vocabulary of Nyāya texts used in new version.

Result:

100% recall.

Remarkable point:

The correct split was always found at the first place.

Performance of two segmenters

Below tables show the difference in performance between SCL-splitter and SCL-NN splitter.

Solutions	Cases	%
0-5	196	55.7
6-10	56	15.9
11-100	72	20.4
101-1000	13	3.6
$>$ 1000	3	1
No Split	12	3.4
Total	352	100

Table : Performance of SCL-splitter

Solutions	Cases	%
1	340	96.59
2	12	3.40
Total	352	100

Table : Performance of modified SCL-NN splitter with controlled lexicon

Grammar of an NNE

The NN expression involves a small number of technical terms together with a non-logical vocabulary(Matilal,1968)

Grammar of an NNE

The NN expression involves a small number of technical terms together with a non-logical vocabulary(Matilal,1968)
Ganeri in the informal description of the NN classifies these into 6 categories.

Grammar of an NNE

The NN expression involves a small number of technical terms together with a non-logical vocabulary(Matilal,1968)
Ganeri in the informal description of the NN classifies these into 6 categories.

- **Primitive Terms**
  These are the nouns such as *ghaṭa* 'pot', *bhūtala* 'ground', *gandha* 'smell', etc..

Grammar of an NNE

The NN expression involves a small number of technical terms together with a non-logical vocabulary(Matilal,1968)
Ganeri in the informal description of the NN classifies these into 6 categories.

- **Primitive Terms**
  These are the nouns such as *ghaṭa* 'pot', *bhūtala* 'ground', *gandha* 'smell', etc..
- **Abstract Functor**
  This is a derivational suffix 'tva' or 'tā' (-ness or -hood), that maps a noun to an abstract noun. For example, *gandha*smell is mapped to *gandhatva* smell-ness, *ghaṭa*pot to *ghaṭatva* pot-ness.

Cont...

- **Relational Abstract Expressions**
  The relational abstract expressions are derived from relation-denoting terms by adding a 'tva' or 'tā' (-ness or -hood) suffix.
  For example, *pitṛ* 'father' is a relation denoting term.
  By adding 'tva' suffix, it changes to *pitṛtva* 'father-hood', a relational abstract expression.
  putratva 'son-hood', *ādheyatā* 'superstratum-ness' and *adhikaraṇatā* 'substratum-ness' are also relational abstract expressions.

Cont...

- **Relational Abstract Expressions**
  The relational abstract expressions are derived from relation-denoting terms by adding a 'tva' or 'tā' (-ness or -hood) suffix.
  For example, *pitṛ* 'father' is a relation denoting term.
  By adding 'tva' suffix, it changes to *pitṛtva* 'father-hood', a relational abstract expression.
  putratva 'son-hood', *ādheyatā* 'superstratum-ness' and *adhikaraṇatā* 'substratum-ness' are also relational abstract expressions.
- **Conditioning Operator**
  The conditioning operator *nirūpita* 'determining' operates on a relational abstract expressions to form a term. For example, *X-nirūpita-pitṛtva* 'father-hood determined by X'.
  Ganeri calls such terms 'relational terms'.

Cont...

- **Sentence-forming Operator**
  The terms such as *niṣṭha* 'resident in' and *avacchinna* 'delimited by' combines a relational term with another term to form a sentence.

Cont...

- **Sentence-forming Operator**
  The terms such as *niṣṭha* 'resident in' and *avacchinna* 'delimited by' combines a relational term with another term to form a sentence.
- **Negation Functor** *abhāvaḥ* 'absence'.

Cont...

- **Sentence-forming Operator**
  The terms such as *niṣṭha* 'resident in' and *avacchinna* 'delimited by' combines a relational term with another term to form a sentence.
- **Negation Functor** *abhāvaḥ* 'absence'.

These 6 categories are necessary to understand both the syntax as well as the semantics of the NNE.

Regrouping the primitives

We regroup these 6 categories into 2 types
viz. the conceptual terms and conceptual relations.

Regrouping the primitives

We regroup these 6 categories into 2 types
viz. the conceptual terms and conceptual relations.

Ganeri's terms	Our terms
Primitive term	Conceptual term
Relational Abstract Expression	-same-
Negation functor	-same-
Abstract functor	-same-
Conditioning Operator	Conceptual Relation
Sentence-forming Operator	-same-

We built a Constituency Parser for NNEs motivated by this grammar.

Constituency Parser of NNEs

A relation in NN is always binary.
The two relata are called *Pratiyogin* and *Anuyogin*.

Constituency Parser of NNEs

A relation in NN is always binary.
The two relata are called *Pratiyogin* and *Anuyogin*.
If 'a' and 'b' are the two relata and 'R' is the relation between them, then the corresponding NN expression has the structure 'a-R-b'.
Here 'a' is the Pratiyogin and 'b' is the Anuyogin.
For example: *Ghaṭa-niṣṭha-ghaṭatvam*.
Here *Ghaṭa* is the Pratiyogin and *ghaṭatva* is the Anuyogin.

Constituency Parser of NNEs

A relation in NN is always binary.
The two relata are called *Pratiyogin* and *Anuyogin*.
If 'a' and 'b' are the two relata and 'R' is the relation between them, then the corresponding NN expression has the structure 'a-R-b'.
Here 'a' is the Pratiyogin and 'b' is the Anuyogin.
For example: *Ghaṭa-niṣṭha-ghaṭatvam*.
Here *Ghaṭa* is the Pratiyogin and *ghaṭatva* is the Anuyogin.
In a NN expression, the Pratiyogin is always to the immediate left of the relation term.
Sanskrit being a head-final language, the modifiers of the Anuyogin 'b', if present, will occupy the space between 'R' and 'b'.

Constituency Parser of NNEs

A relation in NN is always binary.
The two relata are called *Pratiyogin* and *Anuyogin*.
If 'a' and 'b' are the two relata and 'R' is the relation between them, then the corresponding NN expression has the structure 'a-R-b'.
Here 'a' is the Pratiyogin and 'b' is the Anuyogin.
For example: *Ghaṭa-niṣṭha-ghaṭatvam*.
Here *Ghaṭa* is the Pratiyogin and *ghaṭatva* is the Anuyogin.
In a NN expression, the Pratiyogin is always to the immediate left of the relation term.
Sanskrit being a head-final language, the modifiers of the Anuyogin 'b', if present, will occupy the space between 'R' and 'b'.
And hence, *Anuyogin* can be at far off place to the right of relation.

Cont...

In order to determine *Anuyogin* mechanically, we require contextual knowledge.

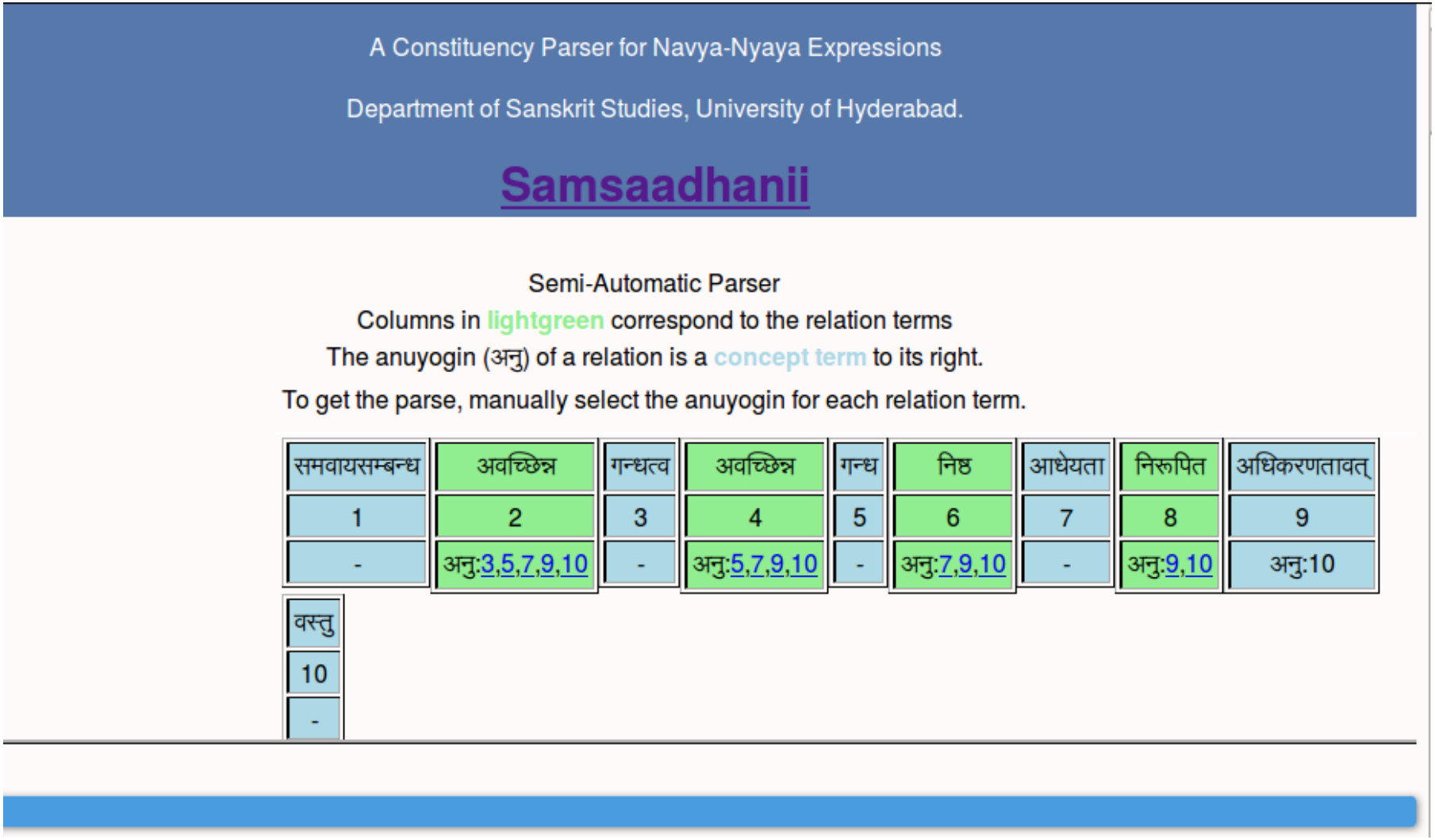
Cont...

In order to determine *Anuyogin* mechanically, we require contextual knowledge.
For instance: *gandhatva*-*avacchinna*-*gandha*-*niṣṭha*-*ādheyatā*

Cont...

In order to determine *Anuyogin* mechanically, we require contextual knowledge.
For instance: *gandhatva-avacchinna-gandha-niṣṭha-ādheyatā*
From this linear presentation, it is not clear whether the anuyogin of avacchinna is gandha or ādheyatā.
It is the contextual knowledge that determines the Anuyogin. We provide an user interface that helps user to select the appropriate Anuyogin.

Cont...


A Constituency Parser for Navya-Nyaya Expressions

Department of Sanskrit Studies, University of Hyderabad.

Samsaadhanii

Semi-Automatic Parser

Columns in lightgreen correspond to the relation terms

The anuyogin (अनु) of a relation is a concept term to its right.

To get the parse, manually select the anuyogin for each relation term.

समवायसम्बन्ध	अवच्छिन्न	गन्धत्व	अवच्छिन्न	गन्ध	निष्ठ	आधेयता	निरूपित	अधिकरणतावत्
1	2	3	4	5	6	7	8	9
-	अनु:3,5,7,9,10	-	अनु:5,7,9,10	-	अनु:7,9,10	-	अनु:9,10	अनु:10

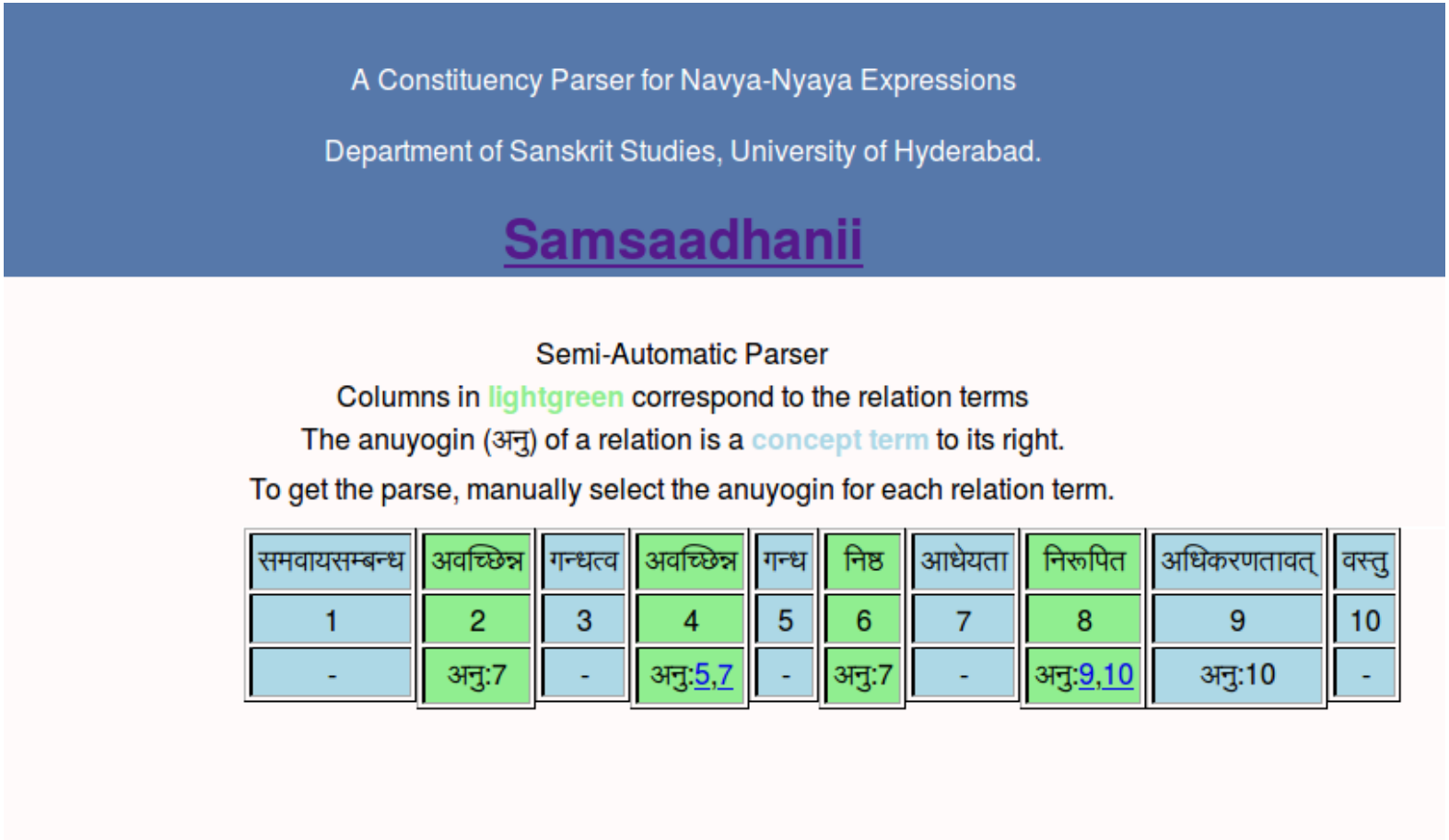
वस्तु
10
-

Figure : A screenshot of interface

Cont...

When the user selects an anuyogin, then the nested parenthesis constraint removes all incompatible solutions reducing the possibilities at each selection.

Cont...



A Constituency Parser for Navya-Nyaya Expressions

Department of Sanskrit Studies, University of Hyderabad.

Samsaadhanii

Semi-Automatic Parser

Columns in lightgreen correspond to the relation terms

The anuyogin (अनु) of a relation is a concept term to its right.

To get the parse, manually select the anuyogin for each relation term.

समवायसम्बन्ध	अवच्छिन्न	गन्धत्व	अवच्छिन्न	गन्ध	निष्ठ	आधेयता	निरूपित	अधिकरणतावत्	वस्तु
1	2	3	4	5	6	7	8	9	10
-	अनु:7	-	अनु:5,7	-	अनु:7	-	अनु:9,10	अनु:10	-

Figure : A screenshot of interface after user-selection

Important cues in building parser

Important cues in building this semi-automatic parser are:

Important cues in building parser

Important cues in building this semi-automatic parser are:

- Concepts and relations alternate in an NNE.
  For instance: *gandhatva-avacchinna-gandha-niṣṭha-ādheyatā*

Important cues in building parser

Important cues in building this semi-automatic parser are:

- Concepts and relations alternate in an NNE.
  For instance: *gandhatva-avacchinna-gandha-niṣṭha-ādheyatā*
- Co-relative terms in NN
  viz. *Anuyogitā* - *Pratiyogitā* etc.

Important cues in building parser

Important cues in building this semi-automatic parser are:

- Concepts and relations alternate in an NNE.
  For instance: *gandhatva-avacchinna-gandha-niṣṭha-ādheyatā*
- Co-relative terms in NN
  viz. *Anuyogitā* - *Pratiyogitā* etc.
- There is well-nested-ness constraint in NNEs,
  viz.

Important cues in building parser

Important cues in building this semi-automatic parser are:

- Concepts and relations alternate in an NNE.
  For instance: *gandhatva-avacchinna-gandha-niṣṭha-ādheyatā*
- Co-relative terms in NN
  viz. *Anuyogitā* - *Pratiyogitā* etc.
- There is well-nested-ness constraint in NNEs,
  viz.
  1. The *pratiyogin* is always to the immediate left of a relation node.

Important cues in building parser

Important cues in building this semi-automatic parser are:

- Concepts and relations alternate in an NNE.
  For instance: *gandhatva-avacchinna-gandha-niṣṭha-ādheyatā*
- Co-relative terms in NN
  viz. *Anuyogitā* - *Pratiyogitā* etc.
- There is well-nested-ness constraint in NNEs,
  viz.
  1. The *pratiyogin* is always to the immediate left of a relation node.
  2. *nirūpita* always connect two co-relative terms.

Important cues in building parser

Important cues in building this semi-automatic parser are:

- Concepts and relations alternate in an NNE.
  For instance: *gandhatva-avacchinna-gandha-niṣṭha-ādheyatā*
- Co-relative terms in NN
  viz. *Anuyogitā* - *Pratiyogitā* etc.
- There is well-nested-ness constraint in NNEs,
  viz.
  1. The *pratiyogin* is always to the immediate left of a relation node.
  2. *nirūpita* always connect two co-relative terms.
  3. the well-nested-ness condition reduce the search space to a considerable degree.

Important cues in building parser

Important cues in building this semi-automatic parser are:

- Concepts and relations alternate in an NNE.
  For instance: *gandhatva-avacchinna-gandha-niṣṭha-ādheyatā*
- Co-relative terms in NN
  viz. *Anuyogitā* - *Pratiyogitā* etc.
- There is well-nested-ness constraint in NNEs,
  viz.
  1. The *pratiyogin* is always to the immediate left of a relation node.
  2. *nirūpita* always connect two co-relative terms.
  3. the well-nested-ness condition reduce the search space to a considerable degree.

We are still looking for points which may help us to parse automatically avoiding user-interaction.

Graphical Representation

Parsed expression will be represented in two graphs - Conceptual Graph and Compressed Conceptual Graph.

Graphical Representation

Parsed expression will be represented in two graphs -
Conceptual Graph and Compressed Conceptual Graph.
The Conceptual Graph(CG) of John Sowa was originally designed as a semantic representation for natural language.

Graphical Representation

Parsed expression will be represented in two graphs -
Conceptual Graph and Compressed Conceptual Graph.
The Conceptual Graph(CG) of John Sowa was originally designed as a semantic representation for natural language.

- Readable and formal for computational purpose.
- Represents both the linguistic structure as well as the knowledge structure.
- CG is so general - Parse trees, Petri nets etc. are special cases of this representation.

Cont...

For instance, “A cat is on a mat” is represented in CG -

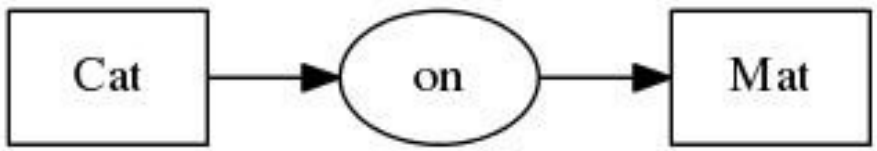


Figure : An example of CG

Here ‘cat’ and ‘mat’ are the concepts and are represented using boxes and the relation ‘on’ is represented using an oval.

Compressed CG

Compressed CG is developed by Amba Kulkarni(1994).

Compressed CG

Compressed CG is developed by Amba Kulkarni(1994).
Here relations are shown by the arrows, not in separate oval nodes.

Compressed CG

Compressed CG is developed by Amba Kulkarni(1994).
Here relations are shown by the arrows, not in separate oval nodes.
For instance, "A cat is on a mat" is represented in Compressed CG -

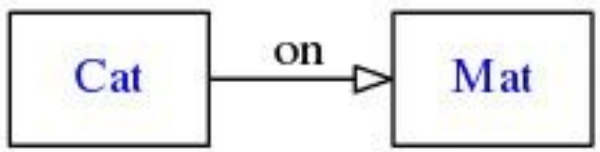


Figure : An example of Compressed CG

Grammar

We wrote a grammar to generate the diagram automatically.

Grammar

We wrote a grammar to generate the diagram automatically.
The grammar is -

```
NNE                : compound_concept
                   ;
compound_concept   : '<' compound_relation '-' concept_term '>'
                   | '<' concept_term '-' concept_term '>'
                   ;
compound_relation  | '<' concept_term '-' relation_term '>'
                   ;
concept_term       : concept
                   | NNE
                   ;
relation_term      : relation
                   ;
```

Concepts and Relations in Navya-Nyāya

Following table shows the concepts and relations of Navya-Nyāya.

Concepts	Nouns Relational Abstract Expressions Abhāva Terms derived with *tva* suffix from nouns
Relations	*niṣṭha*, *avacchinna*, *nirūpita* inverse↓ ↓ ↓realtions *vṛtti/āśraya*, *avacchedaka*, *nirū paka*

Semantics of the Grammar

The attribute grammar defining the synthesized attributes is shown here.

NNE : Compound_Concept
$\uparrow.head = \downarrow.head$
;

Compound_Concept : '<' Compound_Relation '-' Concept_term '>'
$\uparrow.head$ = *Concept_term.head*
establish an edge between the head of the Compound_Relation to the head of the Concept_term
;

Compound_Relation : '<' Concept_term '-' Rel_term '>'
$\uparrow.head$ = *Rel_term.position*
draw a relation node for Rel_term.
establish an edge between the head of the Concept_term to the head of the relation node.
;

Concept_term : NNE
$\uparrow.head = \downarrow.head$
| CONCEPT
$\uparrow.head = \downarrow.position$
draw a concept node
;

Rel_term : relation
$\uparrow.head = \downarrow.head$
;

Figure : Production rules with attributes

An illustration

For instance, <<*gandha-niṣṭha*>-*ādheyatā*>

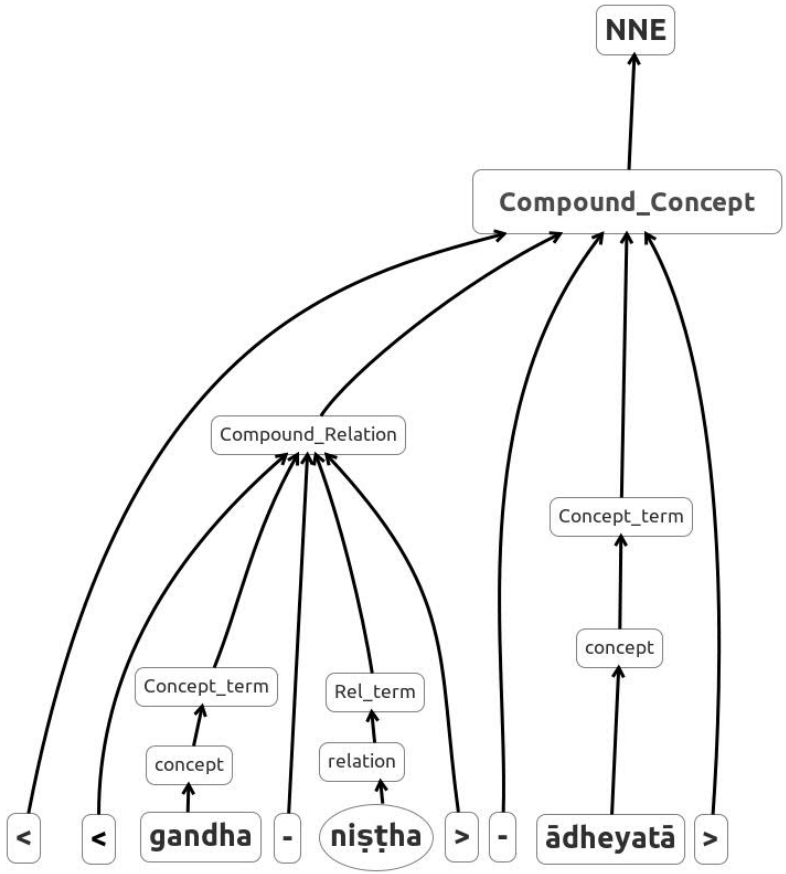


Figure : Constituency parse corresponding to the grammar

Cont...

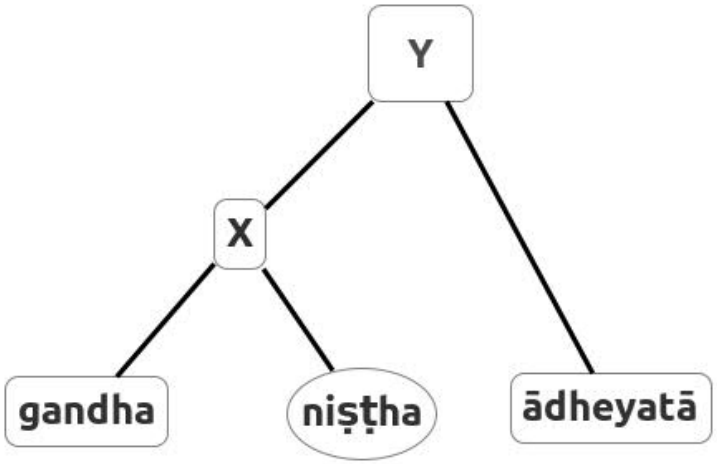


Figure : Compact parse

Cont...

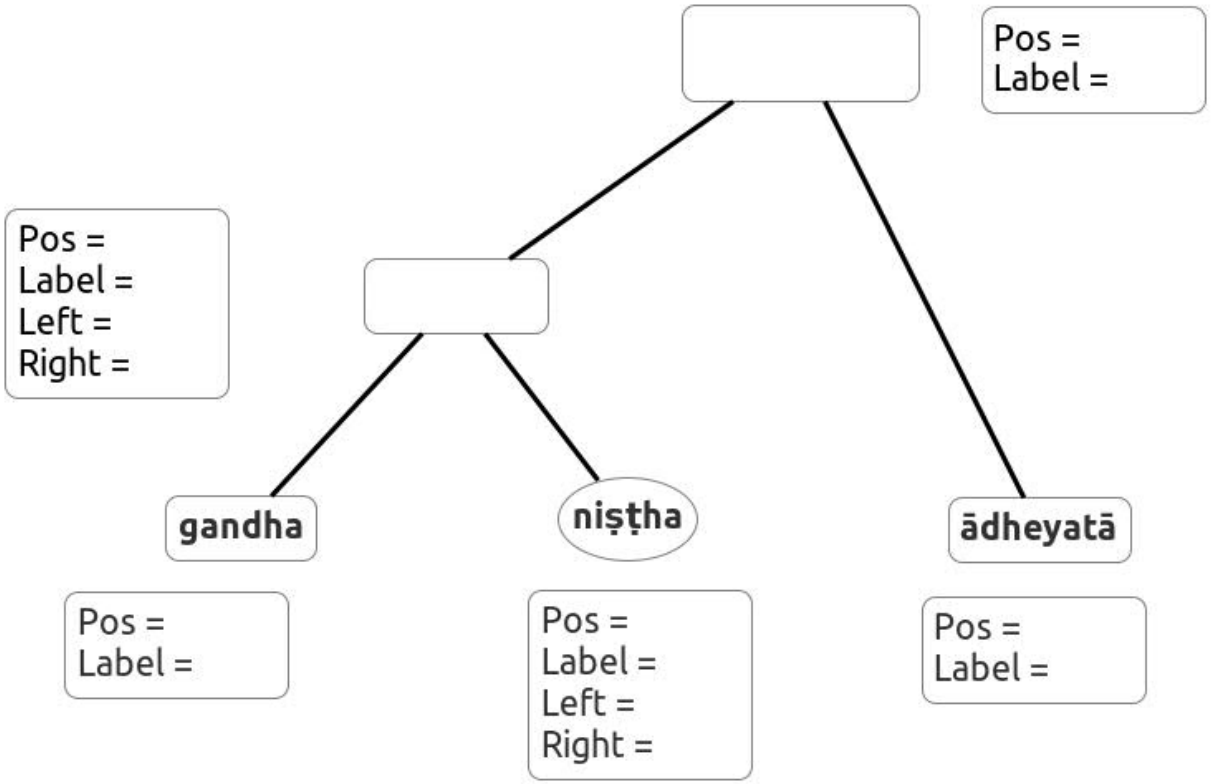


Figure : Concept and relation structure

Cont...

Various stages in Parsing are shown in the Figures 10, 11, 12 and 13. Once every relation node gets its right node position filled in, we draw the CG.

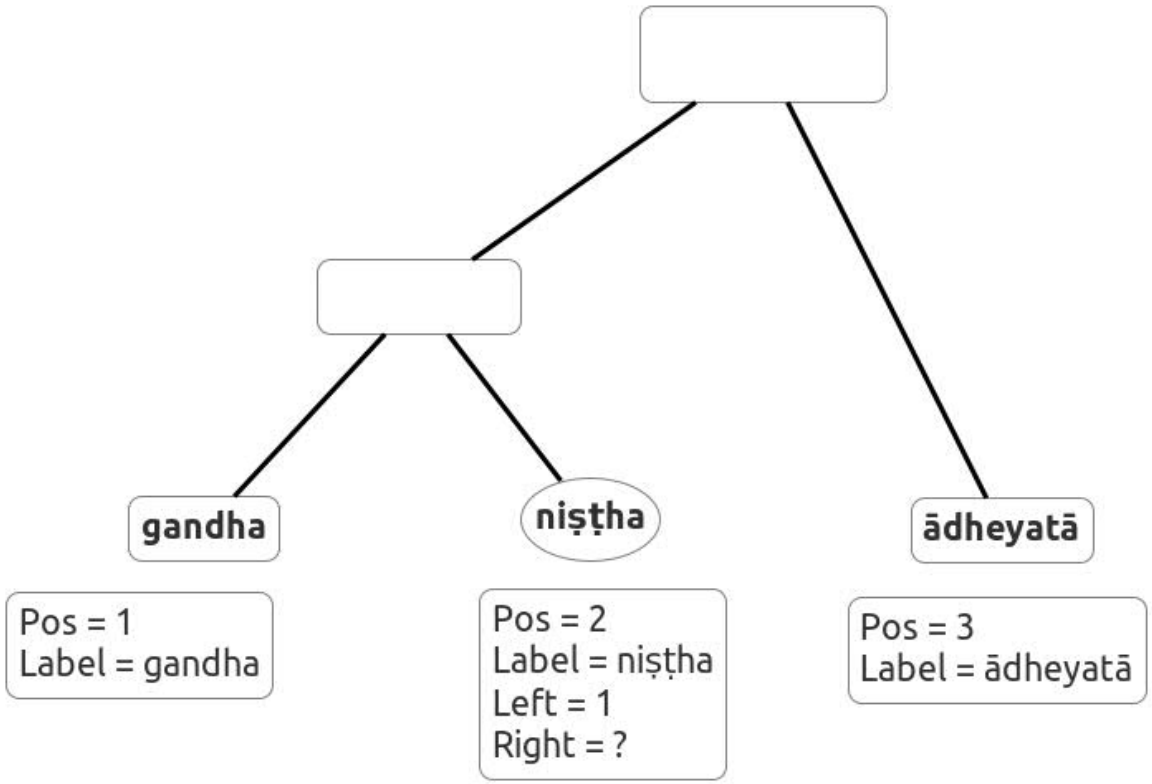


Figure : Intrinsic attributes from lexer

Cont...

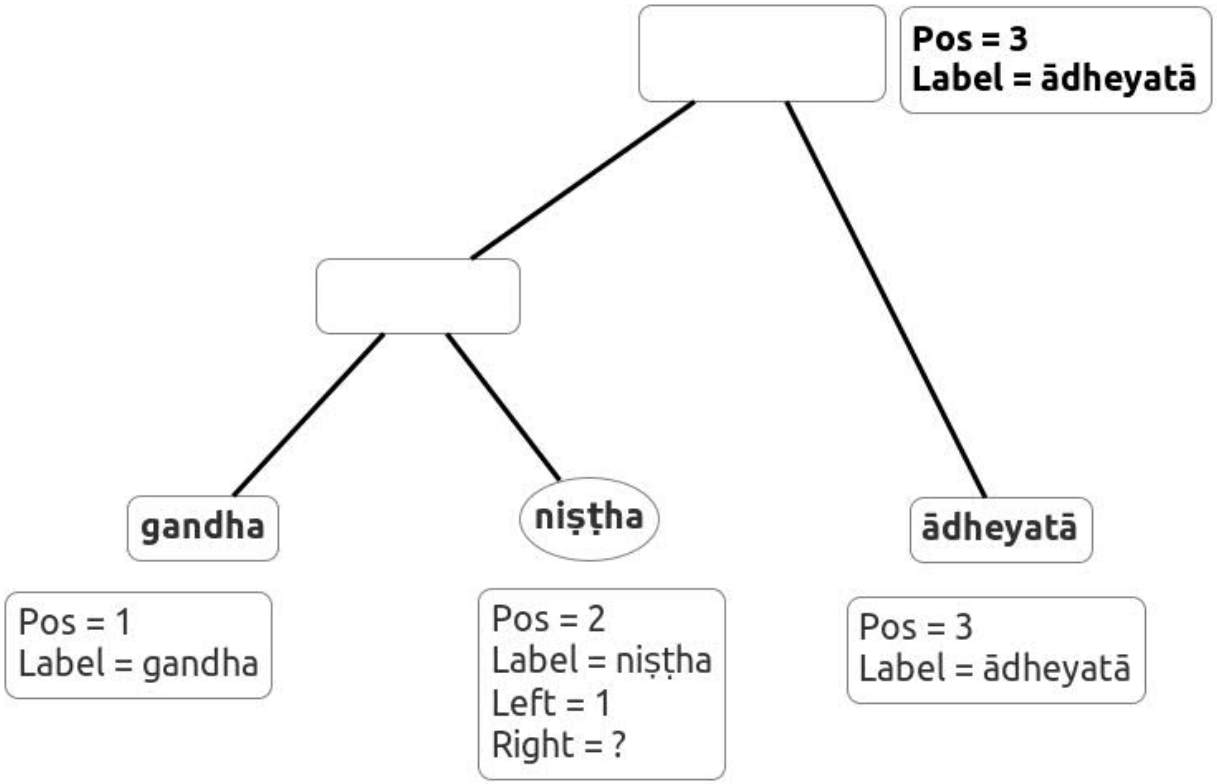


Figure : Compound_Concept acquires features from its child

Cont...

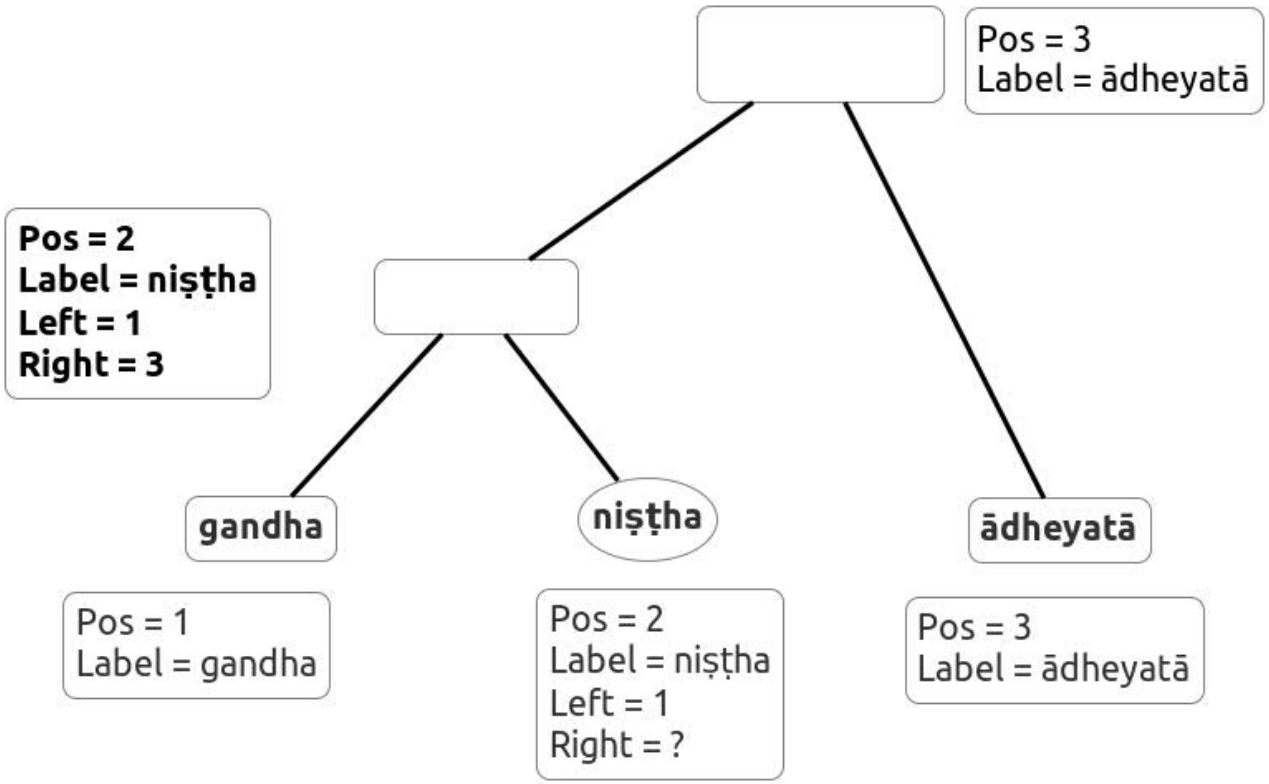


Figure : Compound_Relation inherits 'right' from the parent node and acquires other features from its child

Cont...

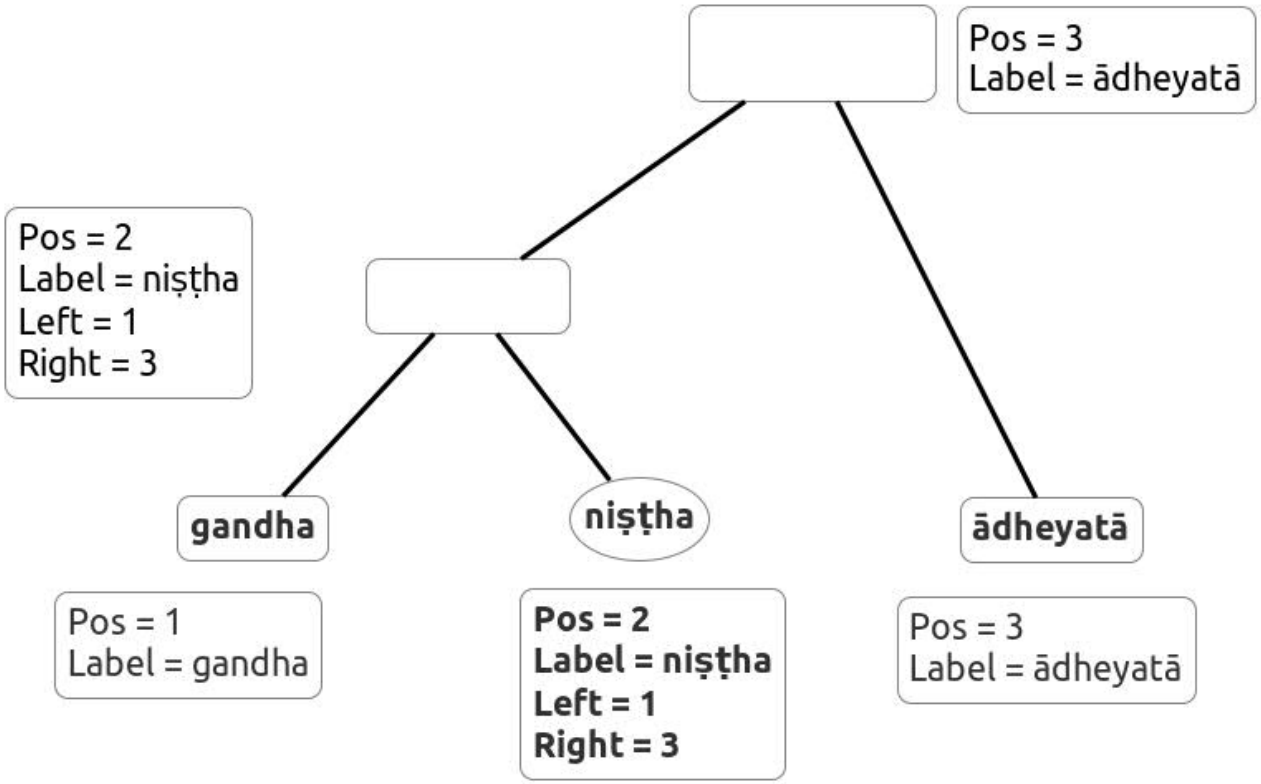


Figure : relation inherits the position of 2^{nd} relata

Nyāyacitradīpikā

Nyāyacitradīpikā (न्यायचित्रदीपिका) is a complete system including all three programs.

Nyāyacitradīpikā

Nyāyacitradīpikā (न्यायचित्रदीपिका) is a complete system including all three programs.
Any NNE can be input for this.

Nyāyacitradīpikā

Nyāyacitradīpikā (न्यायचित्रदीपिका) is a complete system including all three programs.
Any NNE can be input for this.
It segments the given NNE and that becomes an input for Constituency Parser.

Nyāyacitradīpikā

Nyāyacitradīpikā (न्यायचित्रदीपिका) is a complete system including all three programs.
Any NNE can be input for this.
It segments the given NNE and that becomes an input for Constituency Parser.
The correct parse is selected semi-automatically with user-interaction.
The parsed structure is input for Graph renderer.

Demo

DEMO of Nyāyacitradīpikā.

End

Thank you.
Questions/comments are welcome.