From Database to Treebank: Enhancing a Hypertext Grammar with Grammar Engineering

Emily M. Bender
University of Washington

Conference on Electronic Grammaticography
University of Hawai'i
13 Februrary 2011

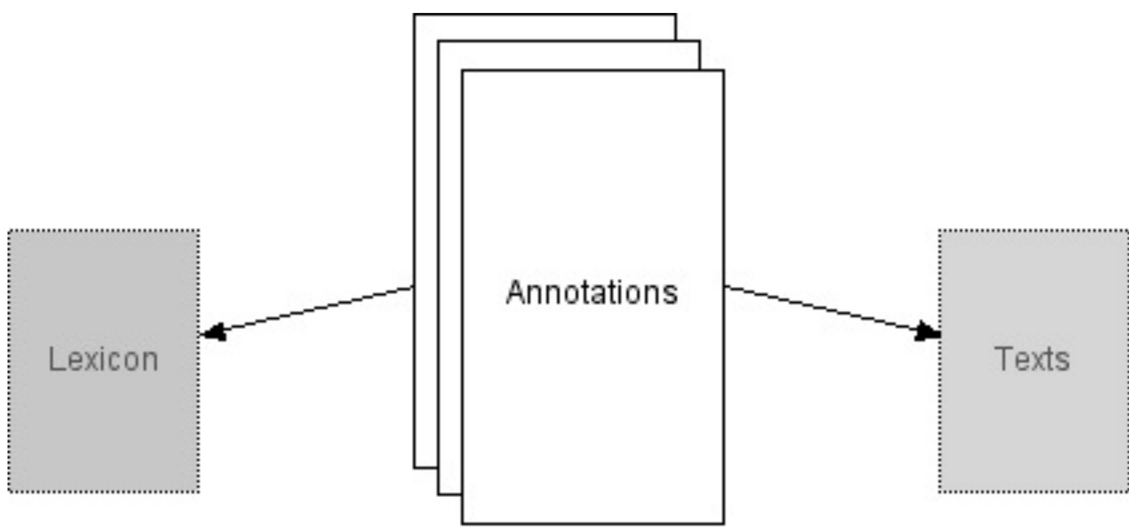
Introduction: Grammatical Descriptions and Implemented Grammars

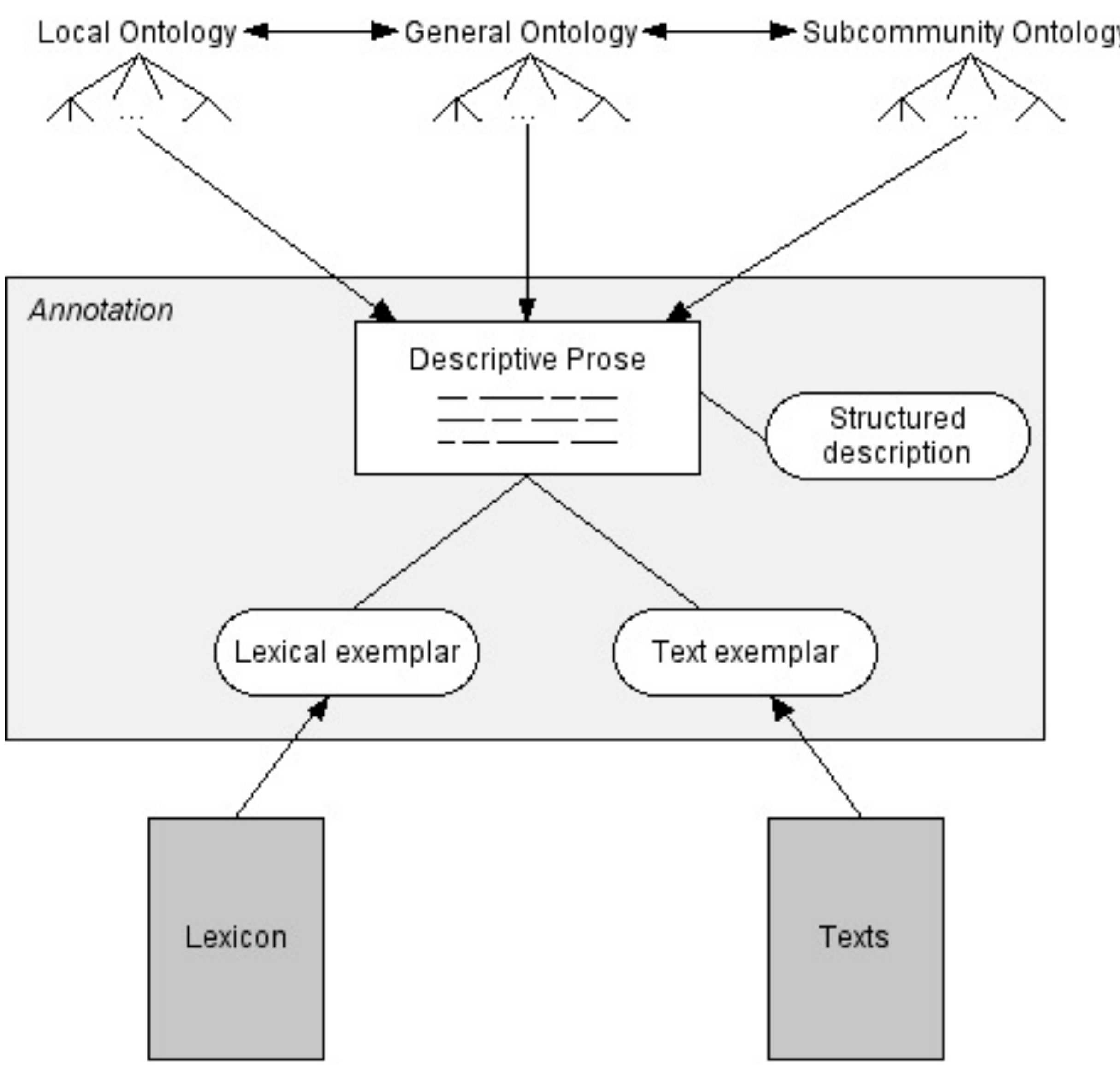
- Good (2004) conceptualizes a descriptive grammar (GD) as a set of annotations over texts and lexicon.
- Annotations take the form of prose descriptions or structured descriptions.
- Annotations are illustrated with exemplars drawn from the text but are understood to express generalizations over more examples.
- Implemented grammars can be understood as machine-readable structured descriptions.
- Those descriptions must be integrated with each other to form a cohesive whole.
- Implemented grammars can automatically produce annotations over individual examples, which can be aggregated and searched.

Overview

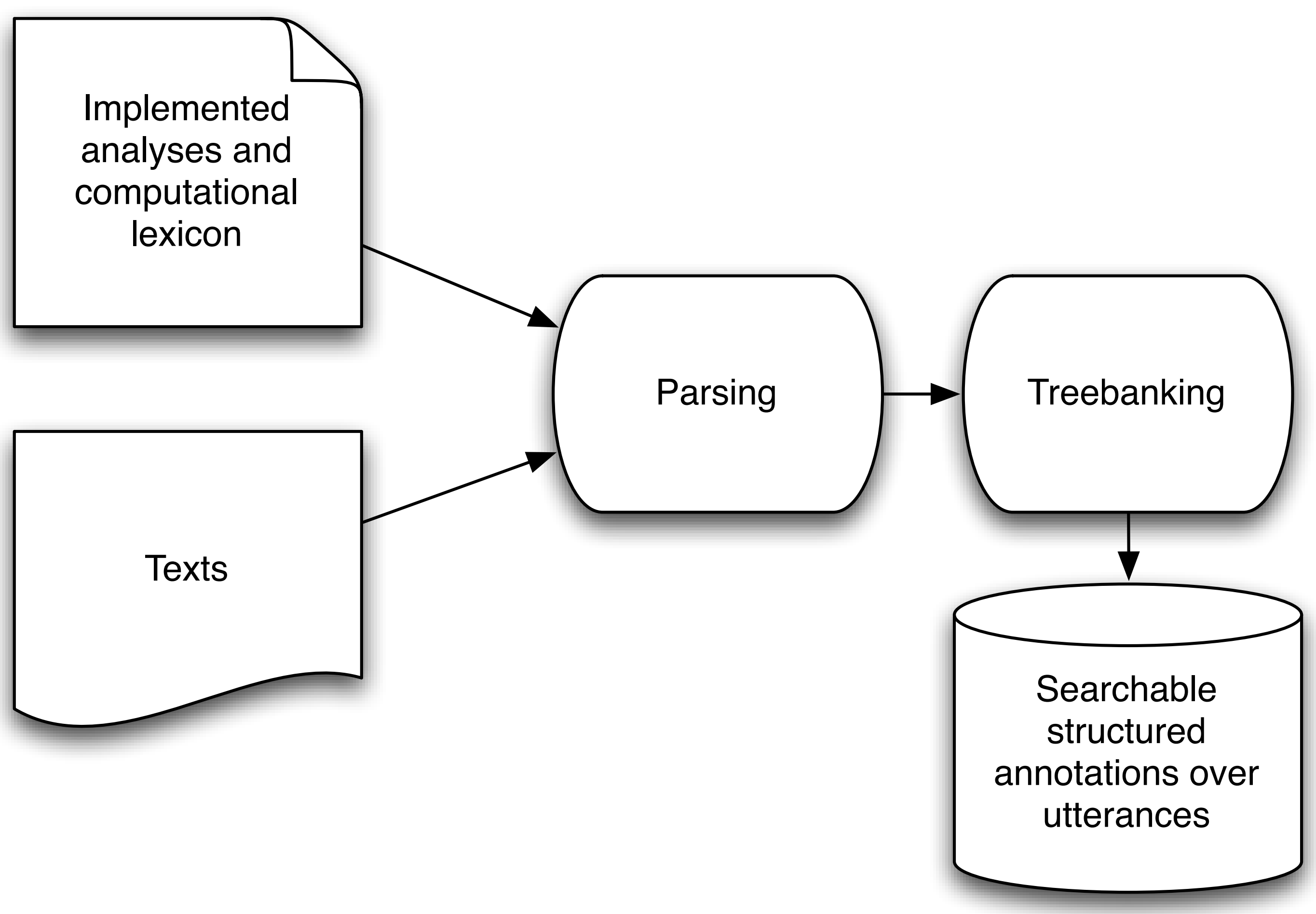
- Introduction
- Implemented Grammars and Treebanks
- Values and Maxims
- Getting There
- Virtuous Cycles and the Montage Vision

In pictures: Grammatical Descriptions (Good 2004)

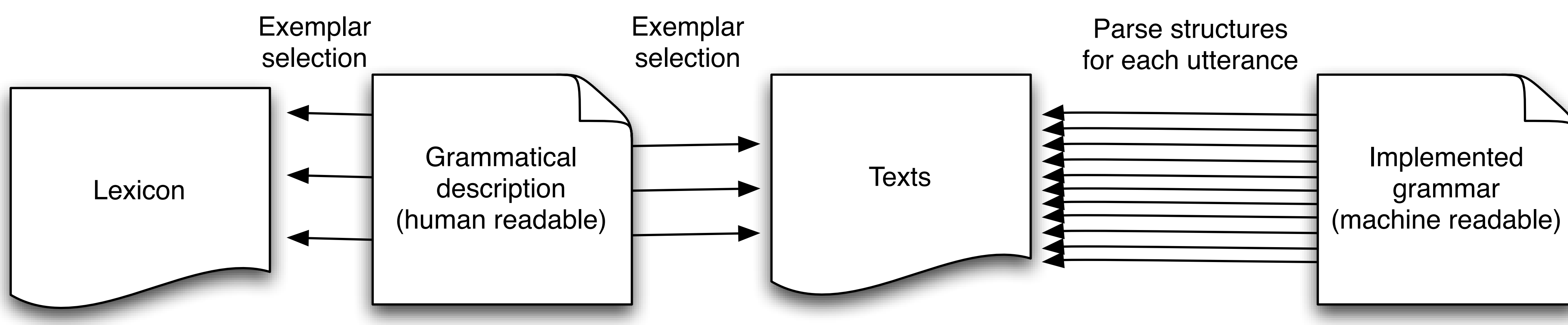




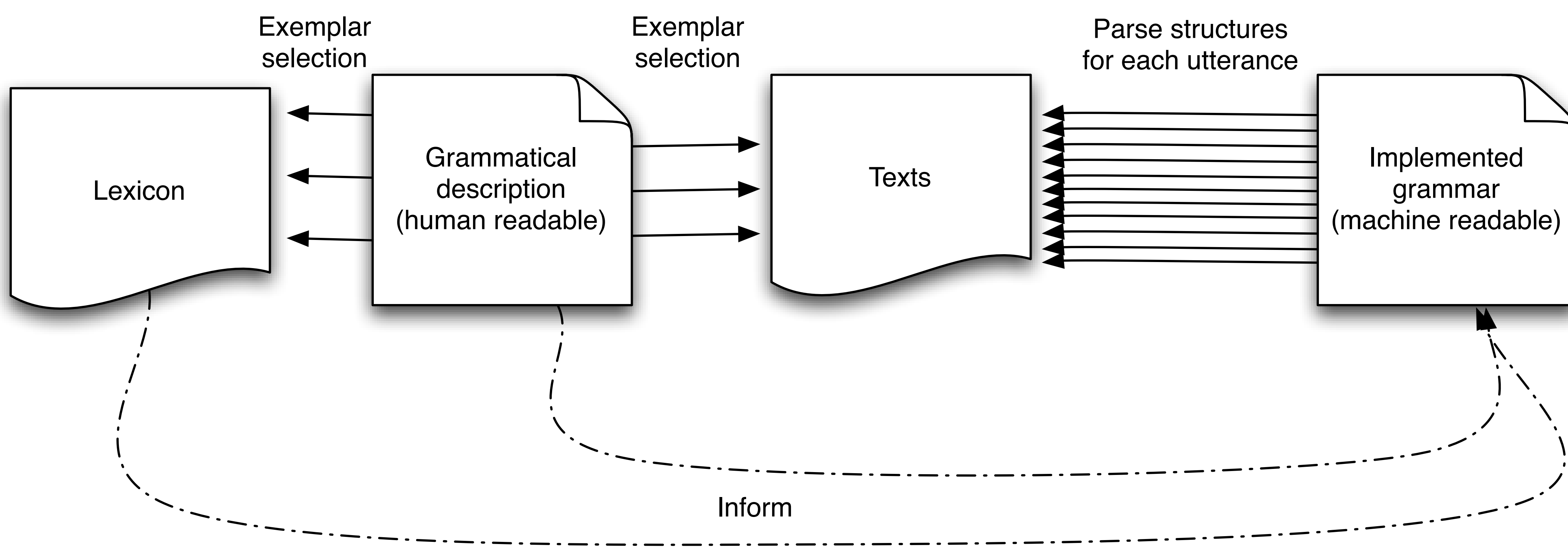
In pictures: Implemented Grammars



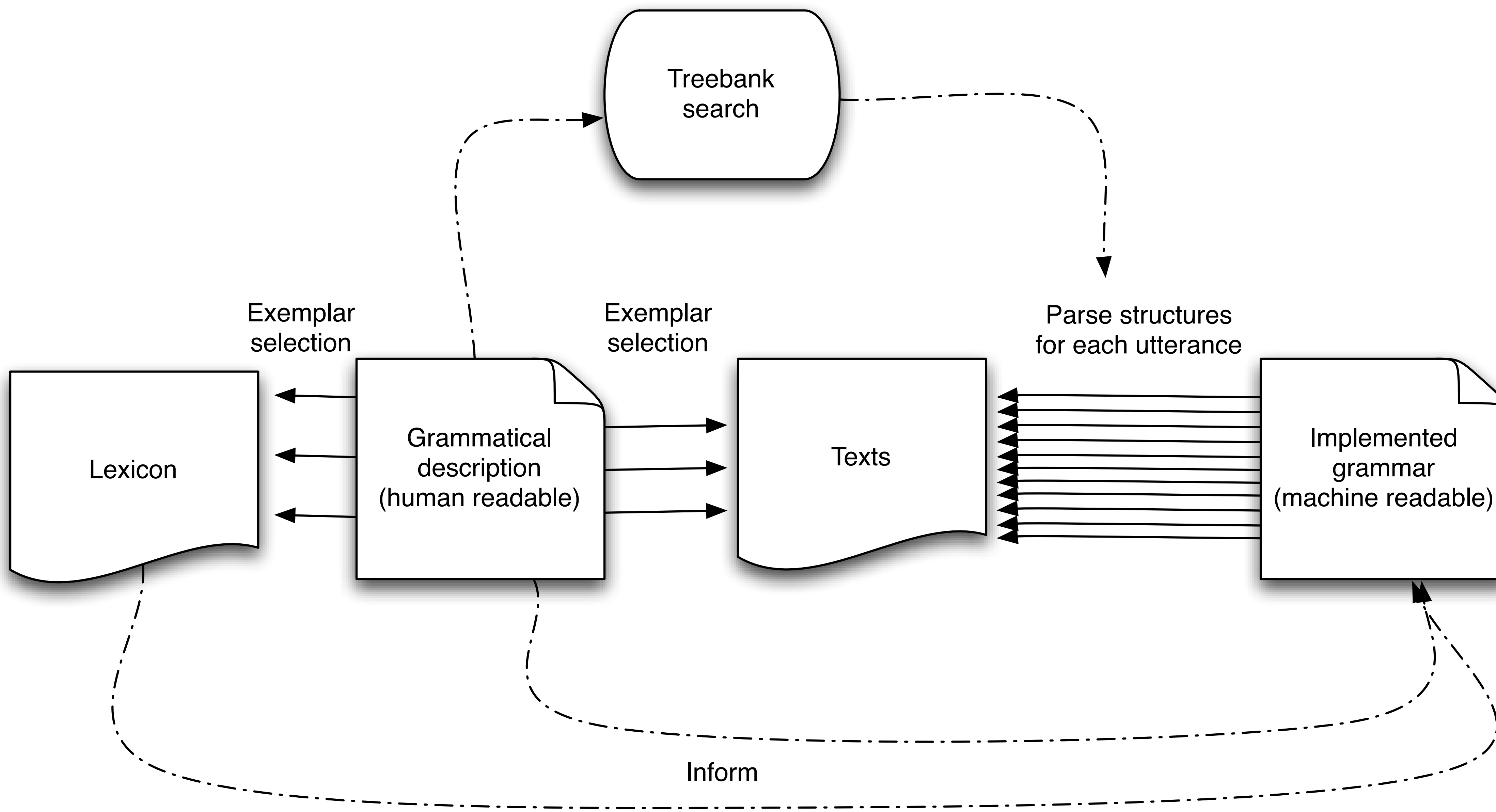
The Big Picture



The Big Picture



The Big Picture



Overview

- Introduction
- Implemented Grammars and Treebanks
- Values and Maxims
- Getting There
- Virtuous Cycles and the Montage Vision

Implemented Grammars

- Comprised of sets of mutually consistent rules and lexical entries
- Make analyses precise enough for a computer to handle them
- Are necessarily *formalized* but are not typically *formalist*
- Currently most developed for syntax, morphology, phonology

Example Grammar: HPSG Grammar of Wambaya (Bender 2008, 2010)

- Based on Nordlinger 1998

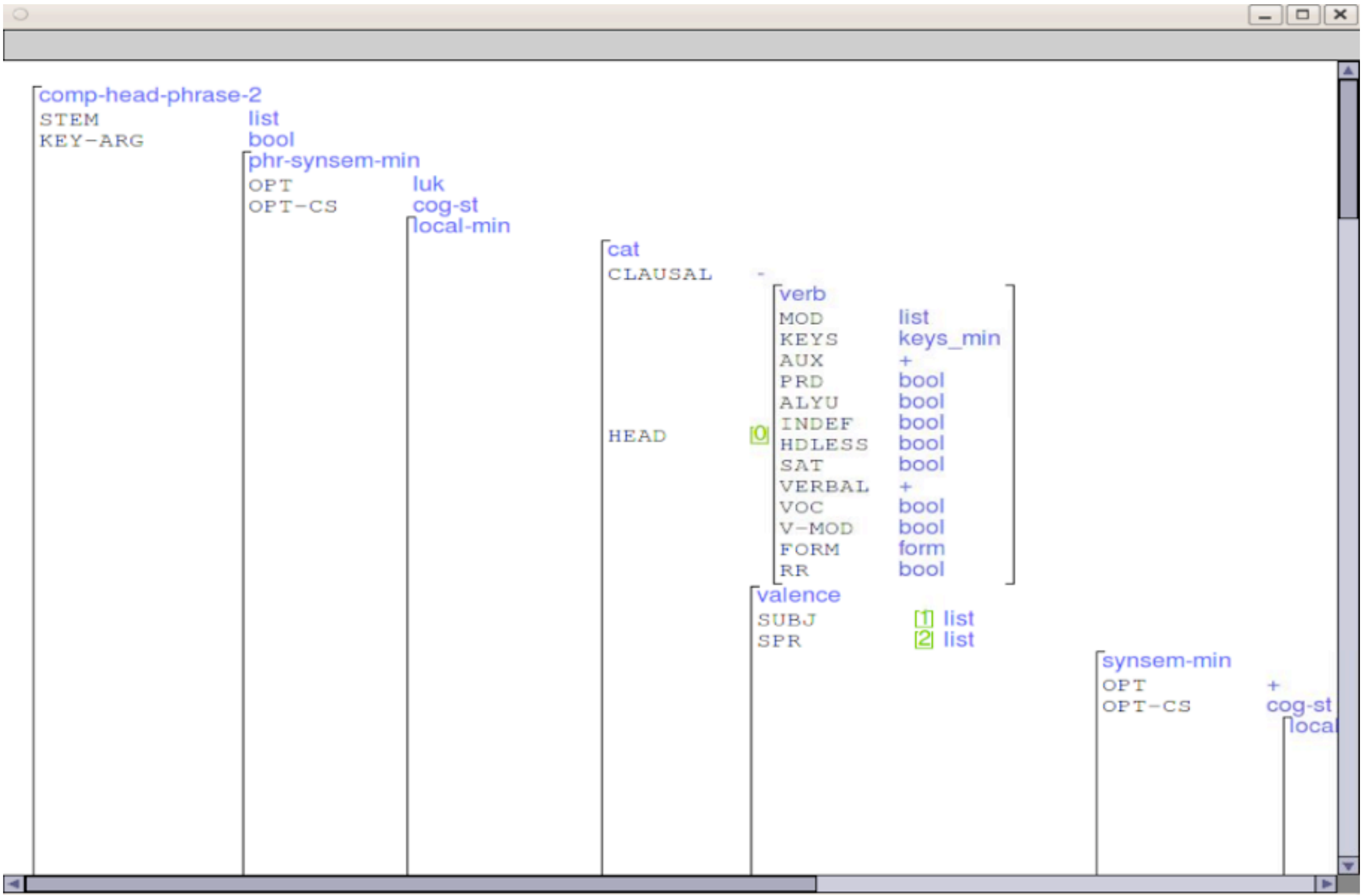
- Developed on the basis of the LinGO Grammar Matrix (Bender et al 2002, 2010)

Definition of a grammar rule

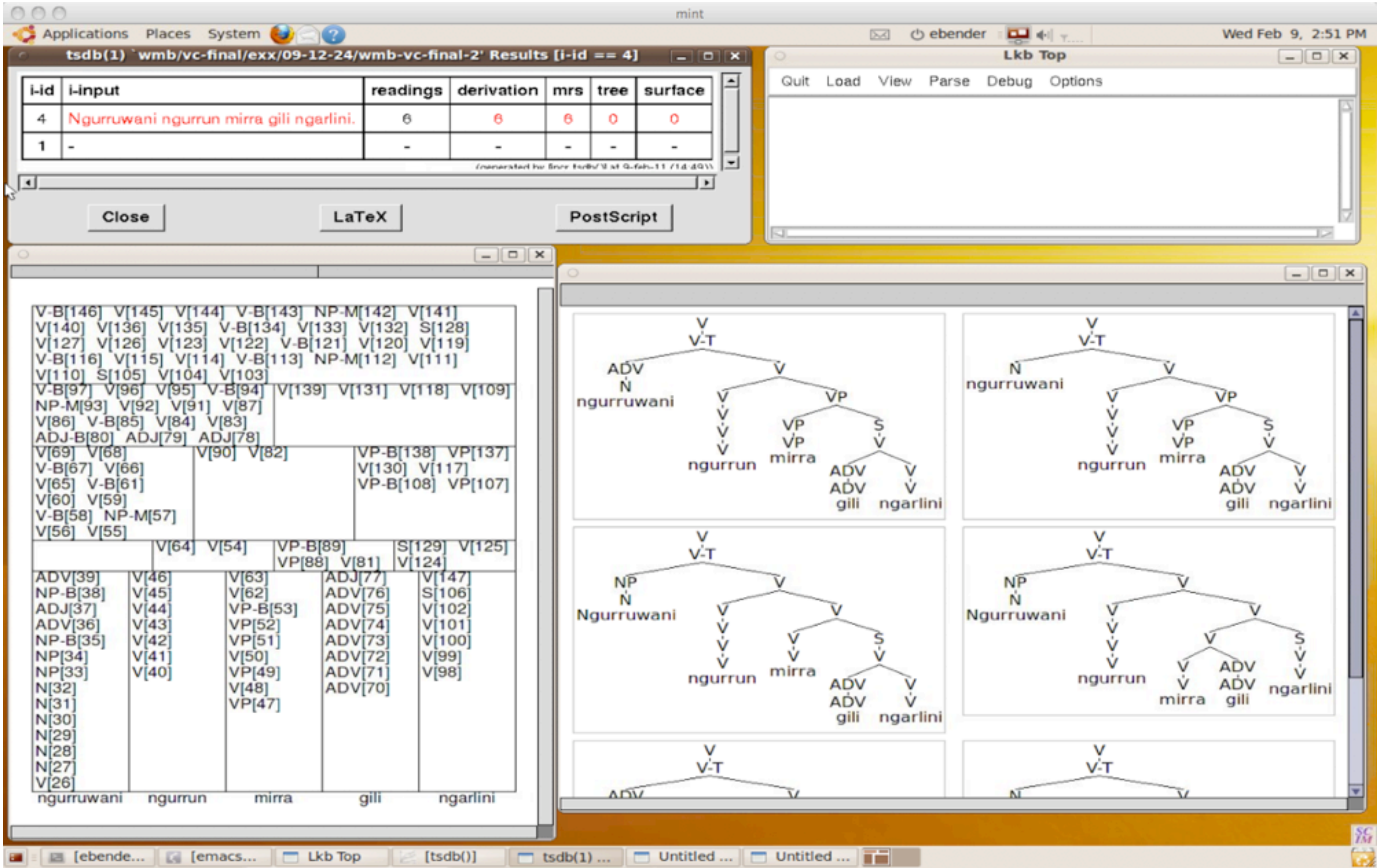
```
wmb-head-2nd-comp-phrase := non-1st-comp-phrase &
  [ SYNSEM.LOCAL.CAT.VAL.COMPS [ FIRST #firstcomp,
             REST [ FIRST [ OPT +,
                  INST +,
                  LOCAL #local,
                  NON-LOCAL #non-local ],
               REST #othercomps ]],
    HEAD-DTR.SYNSEM.LOCAL.CAT.VAL.COMPS [ FIRST #firstcomp,
                 REST [ FIRST #synsem &
                        [ INST -,
                      LOCAL #local,
                      NON-LOCAL #non-local ],
                   REST #othercomps ]],
    NON-HEAD-DTR.SYNSEM #synsem ].

head-comp-phrase-2 := wmb-head-2nd-comp-phrase & head-arg-phrase.
comp-head-phrase-2 := wmb-head-2nd-comp-phrase & verbal-head-final-
                      head-nexus.
```

Inspecting a Grammar Rule



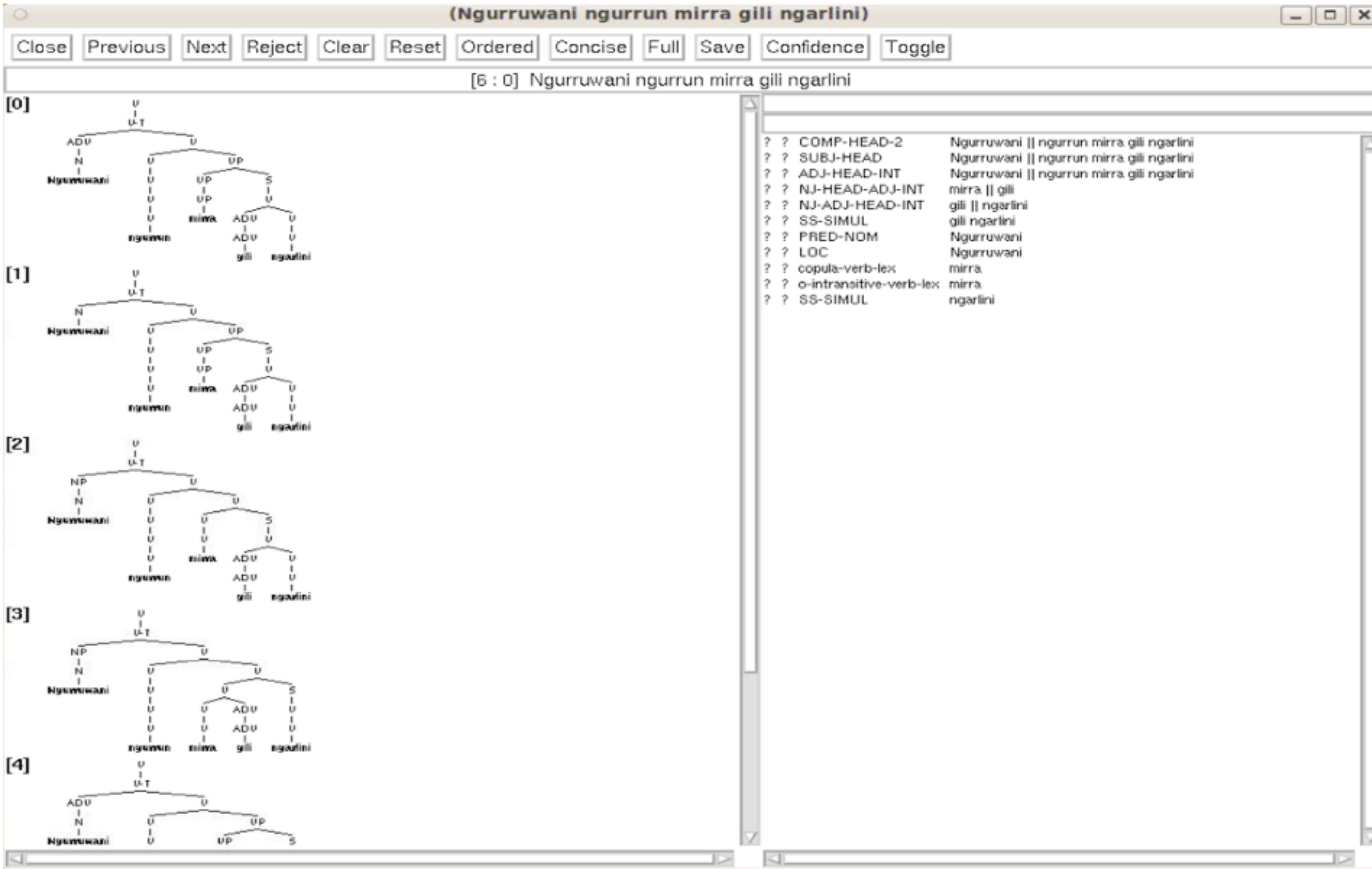
A Grammar Rule in Action



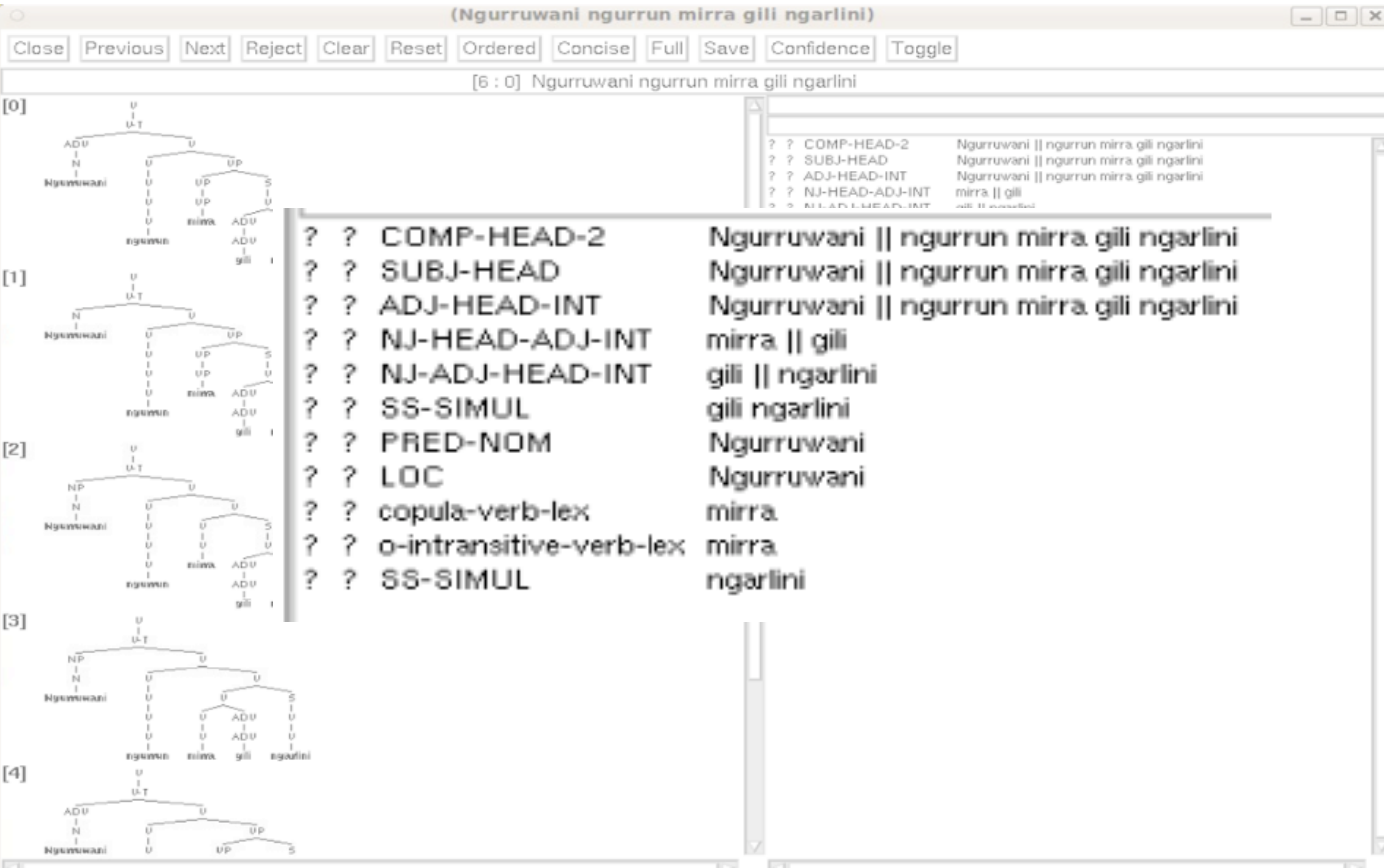
Treebanks

- Old-style (e.g., Penn Treebank, Marcus et al 1993): Develop extensive code book and hand-annotate tree structures for each item.
- New-style (e.g., Redwoods, Oepen et al 2004):
  - Process all items (typically utterances or sentences) with grammar
  - Select intended structure from among those provided by the grammar for each item --- assisted by calculation of discriminants
  - Indicate items with no correct analysis
  - Save decisions to rerun when grammar is updated
- Internally consistent treebanks, which can be updated easily as grammar is improved.

Redwoods Treebanking Tool



Redwoods Treebanking Tool



What Are Treebanks Good For?

- In Computational Linguistics:
  - Training parse-ranking models and other applications of machine learning
- In Language Description:
  - a set of searchable annotations
  - more detailed than IGT
  - more easily kept internally consistent than IGT
  - ... by no means a replacement for IGT!

Treebank Search (Ghodke and Bird 2010)

- Fast queries over large treebanks, including both PTB-style and Redwoods-style

- Sample query over Wambaya data:

  - Find sentences with a complement realized only by a modifier:

```
//DECL[//HEAD-COMP-MOD-2 AND NOT //HEAD-COMP-2
AND NOT //COMP-HEAD-2]
```

  - Find sentences with two overt arguments:

```
//DECL[//J-STRICT-TRANS-VERB-LEX AND
//HEAD-COMP-2 AND //HEAD-SUBJ]
```

Treebank Search

Overview

- Introduction
- Implemented Grammars and Treebanks
- Values and Maxims
- Getting There
- Virtuous Cycles and the Montage Vision

Values and Maxims

- Nordhoff (2008) (following Bird and Simons 2003) presents a series of “values” and “maxims” for electronic GDs.

- The treebanking methodology advocated here speaks to many of these values and associated maxims.

Values and Maxims: Data Quality

- ACCOUNTABILITY: More sources for a phenomenon are better than fewer sources. (Rice 2006:395; Noonan 2006:355; Nordhoff 2008:299)
    - Treebank search helps GD readers turn up examples from texts
- ACTUALITY: A GD should incorporate provisions to incorporate scientific progress. (Nordhoff 2008:299)
    - The Redwoods methodology for producing *dynamic* treebanks ensures that the treebank can always be easily updated when the implemented grammar is.
- HISTORY: The GD should present both historical and contemporary analyses. (Noonan 2006:360; Nordhoff 2008:300)
    - The same software that supports treebanking allows for detailed comparisons between treebanks based on different grammar versions.

Values and Maxims: Exploration

- INDIVIDUAL READING HABITS: A GD should permit the reader to follow his or her own path to explore it. (Nordhoff 2008:303)

    - Major contrast here is form-based versus function-based. In principle, implemented grammars can be used in parsing (string to semantics) and generation (semantics to string)

- EASE OF EXHAUSTIVE PERCEPTION: The readers should be able to know that they have read every page of the grammar. (Nordhoff 2008:305)

    - Problematic for implemented grammars

Values and Maxims: Exploration

- RELATIVE IMPORTANCE: The relative importance of a phenomenon for (a) the language and (b) language typology should be retrievable (Zaefferer 1998c:2; Noonan 2006:355; Nordhoff 2008:306).
  - For a language: Can measure how frequently the constraints associated with that phenomenon appear in the treebank and/or how many grammar components mention them.
  - For typology: Cross-linguistic comparison facilitated by code sharing across implemented grammars.
- QUALITY ASSESSMENT: The quality of a linguistic description should be indicated. (Nordhoff 2008:306)
  - Treebank search can quantify number of examples involving a phenomenon; can be used to estimate coverage of analyses over texts.

Values and Maxims: Exploration

- MULTILINGUALIZIATION: A GD should be available in several languages, among others the language of wider communication of the region where the language is spoken (Weber 2006a:433; Nordhoff 2008:307).
  - Implemented grammars can be used in machine translation. Small MT systems could provide an interesting means of exploration, and one that is fairly easily adapted for different input languages.
- MANIPULATION: The data presented in a GD should be easy to extract and manipulate (Nordhoff 2008:307).
  - Implemented grammars can be used for interactive parsing and generation.

Overview

- Introduction
- Implemented Grammars and Treebanks
- Values and Maxims
- Getting There
- Virtuous Cycles and the Montage Vision

Getting There: Isn't that too much work?

- The original field and descriptive work is the hard part; grammar engineering effort is small in comparison:
  - Bender's (2008) grammar of Wambaya built in 210 hours, or 1/20th the time of the original fieldwork by Nordlinger.
  - 91% treebanked coverage of 804 exemplars in Nordlinger 1998, and 76% treebanked coverage on (short) held-out narrative text.
- Potential for collaboration: field linguist and grammar engineer don't have to be the same person
- Even a grammar with partial coverage can be interesting
- The Grammar Matrix provides a head-start (next slide)

The Grammar Matrix: http://www.delph-in.net/matrix

- A repository of implemented analyses, including:
  - A core grammar with analyses of general patterns such as semantic compositionality
  - “Libraries” of analyses of cross-linguistically variable phenomena
  - Accessible via a web-based questionnaire
  - Produces working HPSG grammars from typological descriptions

Overview

- Introduction
- Implemented Grammars and Treebanks
- Values and Maxims
- Getting There
- Virtuous Cycles and the Montage Vision

Virtuous Cycles and the Montage Vision

- Wambaya experiment involved “post-hoc” grammar engineering
- The process of implemented grammar development always raises questions about the language (no GD is complete)
- Current project: Working on Chintang, in collaboration with Balthasar Bickel et al, who are still actively working with the speaker community
- While a considerable amount of data collection and analysis has to take place before grammar engineering can get off the ground, there is potential for a feedback loop that speeds up (and strengthens) descriptive work.

Montage

- The Montage project (Bender et al 2004) envisioned a software environment which integrated tools for production of IGT, GDs, and implemented grammars.
- The IGT and GD would inform the implemented grammar, and even possibly be input to a system that could automatically create it
- The implemented grammar would feed into IGT and GD development by finding candidate exemplars of each phenomenon.
- Montage was never funded but nonetheless there is progress in the direction of this vision.

Montage: potential components

- Collaborative annotation and GD development environments, including TypeCraft (Beermann & Mihaylov 2009), GALOES (Nordhoff 2007, 2011), and Digital Grammar (Drude 2011).

- The Grammar Matrix customization system (Bender et al 2010)

- Treebank Search (Godhke & Bird 2010)

- Machine learning algorithms that learn typological properties from IGT (e.g., Lewis & Xia 2008)

Conclusions

- Treebanks can complement other kinds of annotations included in electronic grammatical descriptions.
- Technological and methodological advances (including the Grammar Matrix) greatly reduce the cost of producing treebanks.
- The process of creating a treebank can serve to inform and clarify grammatical descriptions.