CLIMB update

Antske Fokkens
VU University Amsterdam

Overview

- Mini introduction of the idea behind CLIMB
- CLIMB (main) evaluation
- CLIMB tools
- CLIMB outlook

CLIMB (recap)

- There is often more than one way to analyze a specific phenomenon
- Analyses interact
- The choices we make when implement a grammar influence the possibilities we have for phenomena handled in the future

CLIMB

- CLIMB proposes to implement analyses in a metagrammar rather than implementing the grammar directly
- Using code generation allows grammar engineers to maintain alternative analyses in parallel until enough evidence is found for an informative decision

Some advantages

- Increased modularity
- Supports multilingual grammar development
- Facilitates capturing dialectal variations
- Write alternative grammars that support different applications

Evaluating CLIMB

- Can CLIMB be used for large scale grammar development? And how does it compare to regular grammar development?
  - What is large scale grammar development?
  - It is not possible to compare two grammar development approaches while keeping all influential factors stable

Evaluating CLIMB

- Implement the phenomena present in the Cheetah core grammar using CLIMB
- Independently defined goal
- Cheetah's core grammar is used as an illustration of making large scale grammar engineering feasible
- Enough similarities for the comparison to possibly lead to an indication of the impact of the results

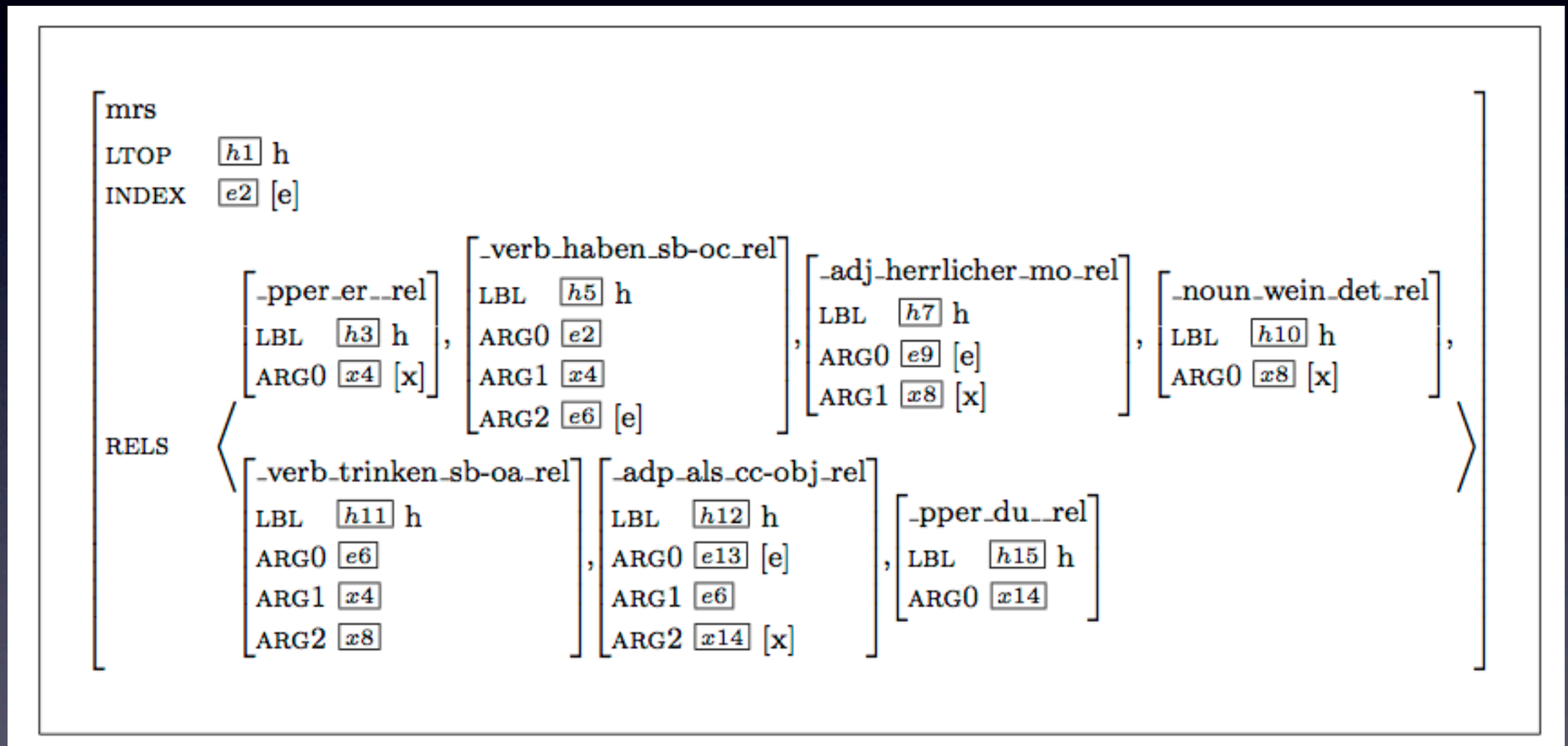
Grammar development

- Cheetah covers 93 examples of a development set containing 106 utterances
- Goal: cover these examples as well
- Using:
  - HPSG literature
  - MRS output of ERG and GG
  - *not* looking at the analyses of Cheetah or GG

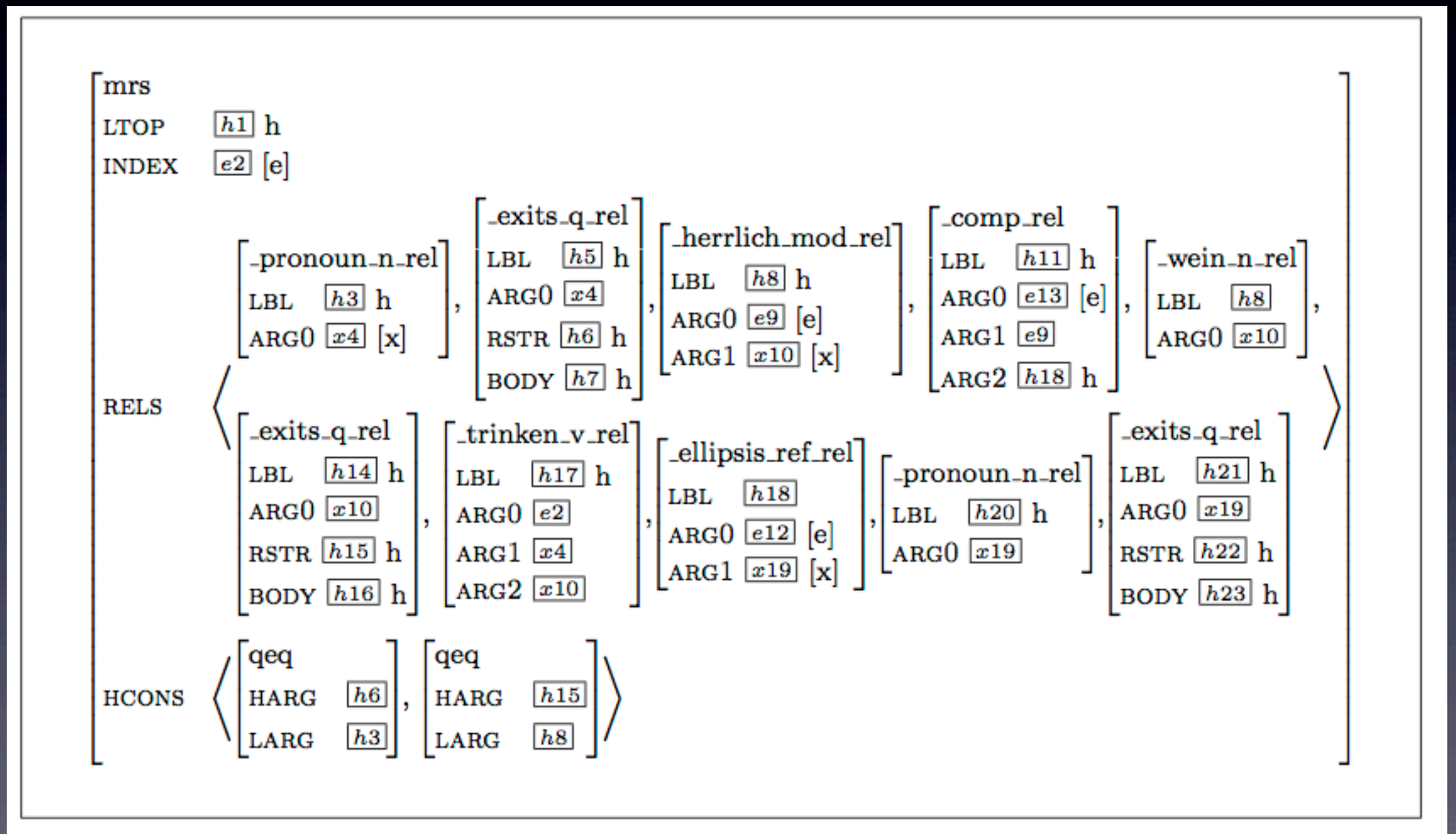
Differences between Cheetah and gCLIMB

- In the early stages, gCLIMB also aimed to cover variations for Dutch and Danish
- Cheetah's MRS representations stay relatively close to syntactic dependencies (for automatic evaluation on TiGer), gCLIMB aims for ERG-like MRSs

Difference in Semantics Cheetah



Difference in Semantics gCLIMB



Outcome (1)

- It took six months to implement gCLIMB grammars that cover the 93 examples covered by Cheetah
- Three main alternative analyses for word order and auxiliary treatment, which cover 5-6 additional examples
- Development for the Cheetah core grammar took 1 year

Outcome (2)

- Both grammars were developed by native Dutch speaking PhD students
- Differences between gCLIMB and Cheetah (attention for other languages and higher demands for MRS output) make development of gCLIMB harder
- It is likely that using CLIMB facilitates grammar engineering and speeds it up

Observations

- So far, the advantage of experience in using CLIMB has out weighted the increase in complexity
- One does not necessarily get to a point where ``conclusive evidence'' for a specific analysis is found

CLIMB tools

- SHORT-CLIMB
- Declarative CLIMB
- Feature geometry extraction, path abbreviation and path completion

SHORT-CLIMB

- Input:
  - File defining changes
  - Complete grammar
- Output:
  - Adapted grammar + (optionally) file to revert changes
  - Reduced grammar + two files to create alternatives

SHORT-CLIMB changes

- := merges constraints to existing type
- :+ merges constraints to existing addendum
- :- removes constraints

- removal=type_name > removes type
- location=type_name + type def > inserts type at location
- complete=on + type def > replaces old definition by new definition (obligatory when changing number of elements on a list)

Declarative CLIMB

- Uses the customization code to generate a ``normal grammar''
- Types can be defined declaratively in TDL
- Properties of types can be made a different locations
- Supports abbreviated paths
- *Choices* indicates which definitions/parts/ files should not be included in the grammar

Declarative CLIMB

- Allows grammar engineers to maintain alternative analyses without becoming procedural programmers
- Does not support the full flexibility of standard CLIMB (morpho-tactics and lexical items)

Feature geometry extraction

- Input:
  - a grammar
- Output:
  - Attributes + types that introduce them + values
  - supertype chains
  - ``default'' values of lists and different lists

Path Abbreviation

- Input:
  - a grammar
  - a feature geometry
- Output:
  - a version of the grammar where all paths are abbreviated as much as possible, i.e. until they can be resolved unambiguously

Path completion

- Input:
  - a grammar that may have abbreviated paths
  - a feature geometry
- Output:
  - a grammar with completed paths, or
  - an error message complaining about an unresolvable path

CLIMB outlook

- Priority:
  - make CLIMB more user friendly
  - how can we combine accessibility of declarative CLIMB with flexibility of standard CLIMB
  - SlaviCLIMB would make an excellent use case to try this out!

gCLIMB outlook

- Recreate gCLIMB in purely Germanic context
- Including analyses from Cheetah and GG
- See if CoreGram analyses can be integrated (can we generate TRALE)
- With user interface and documentation
- Improve TiGer derived lexicon

PLANNING

Coverage of basic German phenomena at the next Delph-In summit would be good

SLOW & STEADY WINS THE RACE