The Chorus/Lisa Prototype(s)

Karsten Konrad, Markus Egg

konrad@cs.uni-sb.de, egg@coli.uni-sb.de

Univ. des Saarlandes

What is Chorus/Lisa?

Ultimate Goal: Understanding Natural Language.

Realistic (?) Goal: A formal system that maps natural language utterances into representations which allow logical reasoning about their meanings.

Chorus/Lisa's Goal: A formal system that maps semantically underspecified (ambiguous, anaphoric, elliptic) utterances into underspecified formal representations which allow disambiguation and anaphoric resolution.

Chorus/Lisa at Present

Chorus: Topic: Disambiguation of underspecified quantifier scope and semantic construction for ellipsis.

Tools: USDL, a Higher-Order semantic representation language, linear HOU and HO-resolution.

Lisa: Topic: Deaccenting, Sloppy Identity, Focus, Parallelism.

Tools: Higher-Order Unification guided by set- and feature-constraints (Colours).

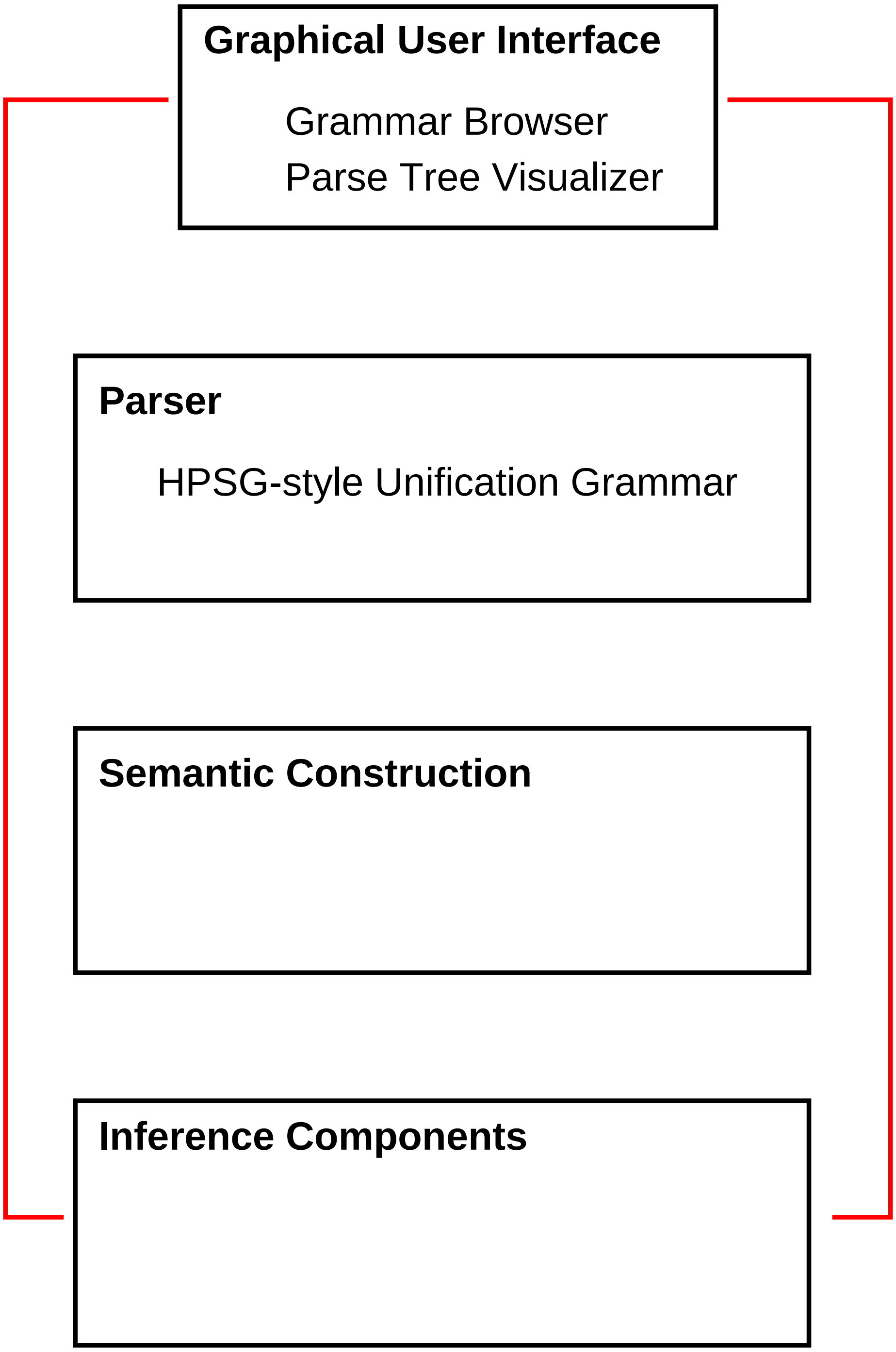
The Chorus/Lisa Prototypes

- Natural Language systems for experimentation with underspecified semantics and other interesting linguistic phenomena

- Final versions should be usable for both research and education

- Systems use shared Code (e.g., parser) and Grammars (syntax, lexicon)

- Implementation in Oz


Graphical User Interface
Grammar Browser
Parse Tree Visualizer
Parser
HPSG-style Unification Grammar
Semantic Construction
Inference Components

LISA

Coloured Lambda-Terms

HO-Coloured Unification
HO-Theorem Proving
Abduction

Chorus

USDL

Linear Unification
Higher-Order Resolution

Why Oz?

Oz is a multiparadigm constraint logic programming language, developed at the Programming Systems Lab in SB, headed by Gerd Smolka.

Feature	Used for
Feature constraints	HPSG-style syntax construction
FO-Unification	Typecheck in λ-calculus
Concurrency	SubCat-Constraints
Encapsulated search	Parser (ambiguous readings)
	HOU (Solutions)
Objects	HO-Unification
Finite-Domain Constraints	Sort hierarchy
Finite-Set Constraints	Colours

NL Semantics and HOU

- **Example**: *John loves his wife. Klaus does too*

- **Analysis**: $\mathit{loves}(\mathit{john}, \mathit{w\text{-}of}(\mathit{john})) \wedge Q(\mathit{klaus})$

- **Equation**: $Q(\mathit{john}) =_{\beta\eta} \mathit{loves}(\mathit{john}, \mathit{w\text{-}of}(\mathit{john}))$

- **Solutions**
  - $Q = \lambda z.\mathit{loves}(z, \mathit{w\text{-}of}(z))$
  * $Q = \lambda z.\mathit{loves}(\mathit{john}, \mathit{w\text{-}of}(z))$
  - $Q = \lambda z.\mathit{loves}(z, \mathit{w\text{-}of}(\mathit{john}))$
  * $Q = \lambda z.\mathit{loves}(\mathit{john}, \mathit{w\text{-}of}(\mathit{john}))$

- **Readings**:
  - Klaus loves his own wife.
  - Klaus loves John's wife.

- The HOU analysis of ellipsis can (correctly) generate the desired ambiguous readings, but it will generally **overgenerate**.

- **Goal:** remove all invalid solutions by formal methods, AS EARLY AS POSSIBLE

Primary Occurrence Restriction

- **Problem**: What linguistic knowledge helps us to find the right solutions?

- **Informal Idea**: [Dalrymple, Shieber, Pereira] Given a labeling of occurrences as either primary or secondary, the POR excludes from the set of linguistically valid solutions, any solution which contains a primary occurrence.

- A **primary occurrence** is an occurrence that is *directly associated* with a *source parallel element*.

- a **source parallel element** is an element of the source (i.e. antecedent) clause which has a parallel counterpart in the target (i.e. elliptic) clause.

- **Example**:
  - $\textit{like}(\underline{John}, \textit{golf}) = Q(\textit{klaus})$
  - $Q = \lambda x.\textit{like}(x, \textit{golf})$
  - $Q = \lambda x.\textit{like}(\underline{\textit{John}}, \textit{golf})$

- The POR describes the syntactic properties of valid solutions, but it does not lead to a suitable algorithm

Coloured λ-calculus

- Developed for inductive theorem proving [Hutter, Kohlhase 1995]

- Symbol occurrences can be annotated with *colours* ($\mathsf{a}, \mathsf{b}, \ldots$), or *colour variables* ($\mathsf{A}, \mathsf{B}, \ldots$)

- Bound variables are uncoloured (coloured $\beta\eta$-conversion just as usual)

- Well-coloured substitutions σ

  - Map X_{a} and c_{A} to formulae of appropriate type

  - $|\sigma(x_{\mathsf{a}})| = |\sigma(x_{\mathsf{b}})|$, i.e. the colour erasures have to be equal.

  - If c is a colour constant, then $\sigma(x_{\mathsf{c}})$ is c-compatible (monochrome).

HOCU: Coloured HOU

- Bindings have to be well-coloured:
  - a-compatible for a colour constant a
  - most general for a colour variable.
- They are essentially unique, given a type, head and colour.
- Given these and coloured simplification, HOCU works exactly as HOU.
- Instead of symbolic colours, we use partial set descriptions (finite set constraints) of Oz 3.0.1

An Example

- **Example**: *Peter says he sees mary and bill does too.*

- **Analysis**: $\mathit{say}(\mathit{peter}, \mathit{see}(\mathit{peter}, \mathit{mary}) \wedge Q(\mathit{bill})$

- **Equation**:
  $Q(\mathit{peter}) =_{\beta\eta}$
  $\mathit{say}(\mathit{peter}, \mathit{see}(\mathit{peter}, \mathit{mary}))$

- **Coloured Equation**:
  $Q_{[pe-]}(\mathit{peter}_{[pe+]}) =_{\beta\eta}$
  $\mathit{say}(\mathit{peter}_{[pe+]}, \mathit{see}(\mathit{peter}_{[pe-]}, \mathit{mary}))$

Conclusion and Future Work

We believe that the Chorus/Lisa-System will become a generally useful workbench for experimentation with underspecified semantics and HO-based semantic construction.

Future work will concentrate on inference components like HO-resolution and an HO-abductive theorem prover.