Linguistic phenomena: phenomena catalogue, how to divide facts up into phenomena in customization-based grammar engineering (CLIMB), phenomena-oriented rather than type-oriented development tools

Delph-in Participants Suquamish,
slides by Antske Fokkens based on notes from Rebecca Dridan

June 27, 2012

The problem:

- Analyses of phenomena typically touch many different types/rules, and each type/rule typically bears constraints related to multiple phenomena.
- Grammar engineers and others might have many reasons to want to see where in a grammar a phenomenon is handled:
  - Learning from someone else's grammar
  - Cross-linguistic comparison: How does phenomenon X differ across grammars?
  - Cross-grammar comparison: How many phenomena are handled in grammars X, Y and Z?
  - Measuring degree of interaction between phenomena
  - Discoverability: How can I find grammars that have treatments of phenomenon Z?
  - Grammar engineering for language documentation ($\Rightarrow$ finding phenomena in a treebank)

A solution:

- Annotate grammars with labels on constraints (i.e., pieces of types) indexing the analyses they belong to
- Ideally using a vocabulary drawn from GOLD or a similar ontology (for discoverability)

But how?

- Build on lextype DB?
- Embedded in tdl or stand-off?
- Grammar Matrix customization system could auto-generate for library-based analyses (and to get people off on a good start)
- Retrofitting might be harder than building in in new grammars

Last year's Discussion (1)

`http://moin.delph-in.net/SuquamishGrammarIndexing`

- Phenomena hard to define, but we can take a practical approach
  - Everyone write down 100 important phenomena (no implementation, not necessarily with name)
  - Annotate MRS test suite with intended phenomena, adapt translations accordingly
- Relating phenomena to parts of the grammar:
  - Use sentences with phenomena to index or find them
  - Label lexical types and MRS with GOLD and use sentences to index phenomena, documentation separated from grammar
    $\rightarrow$ too stand-off: information on types and constraints is needed

Last Year's Discussion (2)

- Use Grammar Matrix to inform interactions, customization to produce documentation
- Use addenda (`:+`) to build types line by line
- If possible document in one place: not all places touched by the grammar
- Tools to show this stuff: code-refactoring?

Conclusion:

Grammar Engineers get phenomena, use type addenda to group features by the phenomenon (not only by type). Editor with layers ala photoshop to deal with this: oe and Mike make this tool?

Today's discussion

- Progress:
  - Lists of phenomena/examples, MRS test suite adaptation?
  - Tools to spot phenomena in grammars
- Next steps:
  - Continue work on phenomena
  - Metagrammar to group analyses
    $\longrightarrow$ Like the addendum proposal but without inconvenience of poor grammar readability
    $\longrightarrow$ If well organized: good for new and small grammars, but how about larger existing grammars?
- Other/new proposals?
- Catalogue of phenomena and implementations