Implementing HPSG
Why implement grammars?

- Practical applications:
  - Machine translation (symbolic, hybrid)
  - Automated customer service response
  - Speech prostheses
  - Hybrid Q&A systems (DeepThought)
  - Language tutors
Why implement grammars?

• Linguistic hypothesis testing:
  • Particular analyses
  • Interaction of analyses
  • Against test suites or corpora
  • Crosslinguistic testing of formalism choices
  • Crosslinguistic testing of proposed universals
Bottom Line

- People are better than computers at language
- BUT: Computers are better than people at keeping track of the details of formal analyses of language
- Further: The precision and coverage required by implementation keeps theoretical work grounded in actual data.
Resources: Delph-In

• LKB: grammar development environment (including parser and generator)
• PET: industrial strength parser
• [incr tsdb()]: Competence and performance profiling laboratory (regression testing)
• Redwoods: Software for creating dynamic, grammar-associated treebanks
Parsing Complexity

• Depends on:
  • Formalism
  • Grammar
  • Sentence length
  • Lexical ambiguity
  • Best first v. exhaustive
How fast is the fast parser?

Table 2: Reference data sets used in comparison and benchmarking with the LinGO grammar.

<table>
<thead>
<tr>
<th>Set</th>
<th>Aggregate</th>
<th>total items</th>
<th>word string</th>
<th>lexical entries</th>
<th>total results</th>
<th>parser analyses</th>
<th>passive edges</th>
<th>fs size</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘tsnlp’</td>
<td>wellformed</td>
<td>1574</td>
<td>4.96</td>
<td>13.8</td>
<td>945</td>
<td>2.00</td>
<td>86</td>
<td>273</td>
</tr>
<tr>
<td></td>
<td>illformed</td>
<td>2775</td>
<td>4.50</td>
<td>11.5</td>
<td>409</td>
<td>1.83</td>
<td>39</td>
<td>257</td>
</tr>
<tr>
<td>‘cslit’</td>
<td>wellformed</td>
<td>918</td>
<td>6.45</td>
<td>15.3</td>
<td>732</td>
<td>2.16</td>
<td>115</td>
<td>302</td>
</tr>
<tr>
<td></td>
<td>illformed</td>
<td>375</td>
<td>6.11</td>
<td>14.9</td>
<td>85</td>
<td>2.31</td>
<td>84</td>
<td>298</td>
</tr>
<tr>
<td>‘aged’</td>
<td>wellformed</td>
<td>96</td>
<td>8.41</td>
<td>23.1</td>
<td>72</td>
<td>7.00</td>
<td>292</td>
<td>315</td>
</tr>
<tr>
<td></td>
<td>illformed</td>
<td>142</td>
<td>11.05</td>
<td>34.2</td>
<td>24</td>
<td>20.33</td>
<td>611</td>
<td>317</td>
</tr>
<tr>
<td>‘blend’</td>
<td>wellformed</td>
<td>1910</td>
<td>11.13</td>
<td>32.1</td>
<td>1008</td>
<td>51.39</td>
<td>1181</td>
<td>336</td>
</tr>
<tr>
<td></td>
<td>illformed</td>
<td>142</td>
<td>11.05</td>
<td>34.2</td>
<td>24</td>
<td>20.33</td>
<td>611</td>
<td>317</td>
</tr>
</tbody>
</table>

Table 6: Development of salient performance parameters (PAGE vs. PET) over three years.
Resources: Delph-In

- Resource grammars: English, German, Japanese, Norwegian, Modern Greek...
- The Grammar Matrix: Rapid-prototyping of scaleable precision grammars
- And more: Delph-In: www.delph-in.net
Related Work

- TRALE: Another grammar development environment for HPSG
- ParGram: Parallel Grammar development for multiple languages in LFG
Grammar Matrix: Motivation

- ERG stats (5/2005)
  - 140,000 lines of code (25,000 exclusive of lexicon)
  - ~3000 types
  - 16+ person-years of effort
- How much of that is useful in other languages?
- How much faster can we develop the next grammar?
Grammar Matrix: Motivation

• Promote consistent semantic representations
• Reuse ‘downstream’ technology in NLU/NLG applications
• Transfer-based MT (symbolic or stochastic)
• Crosslinguistic hypothesis testing
• Endangered language documentation
Matrix: Proposed Universals

• Words and phrases combine to make larger phrases.

• The semantics of a phrase is determined by the meaning of its parts and how they’re put together.

• Some rules for phrases add semantics.

• No rule can remove semantic information.
Matrix: Proposed Universals

• Most phrases have a head daughter.
• Heads determine the types of arguments they require, and how they combine semantically with those arguments.
• Modifiers determine the types of heads they modify, and how they combine semantically with those heads.
Matrix: Proposed Universals

- All NPs are associated semantically with a quantifier.
- Quantification over events is different from quantification over individuals.
- Definiteness/discourse status is expressed in the same way as distinctions between quantifiers.
Matrix: Modules

• Plenty of patterns recur without being universal.

• Current work: Development of ‘modules’ representing different strategies for: word order, sentential negation, coordination, question formation...

• Challenges: Modules aren’t all that ‘modular’
Example: Word order

- The present word order modules constrain:
  - Order of daughters in head-subject and head-complement rules
  - Order of application of head-subject and head-complement rules

- Potentially interacts with:
  - Clause type, part of speech type (of head or dependent), finiteness, ...
Matrix Modules

• Each piece carefully designed to account for one particular aspect of a language...

• ... while interacting appropriately with other modules.

• How best to organize/maintain the various pieces in such a resource?

• How best to ask linguists for the information which will guide module choices?
Language

e.g., Basque or Hawaiian_Creole:

---

Word Order

Please indicate which pattern best describes the basic word order of your language:

- a) SOV
- b) SVO
- c) VSO
- d) OSV
- e) OVS
- f) VOS
- g) V-final
- h) V-initial
- i) free (pragmatically determined) word order

Note: Modules for V2 order (auxiliary second, all else free or finite verb second, non-finite verb clause-finally) and differing word order between matrix and subordinate clauses are currently under development.

Does your language have determiners (as independent words)?

- yes
- no

If so, what is the order of determiners with respect to nouns?

- Noun-Det
- Det-Noun

---

Basic Lexicon

Please use full form lexical entries (not stem forms or citation forms). This form can only accept lower ascii alphanumeric (a-z, A-Z, 0-9, _).

Noun 1:

- Spelling: 

- Predicate name (e.g., _cat_n_rel): 

- For this noun, a specifier (determiner) is:
  
- obligatory
- optional
- impossible
Working with the LKB

Sentence: the cat chased the dog.

Parse Chart for "the..."
Lab: Call for Participation

• Visit the course website for lab instructions
• Software is available on the PCs on the computer classroom, or for download to your own computer
• I’ll be checking the bulletin board frequently!