The Grammar Matrix

Scott Drellishak
University of Washington

August 20, 2007

1

[1] This work is supported by NSF grant BCS-0644097, a gift to the Turing Center by the Utilika Foundation, and the Max Planck Institute for Evolutionary Anthropology.

Current Work

- Grammar Engineering class taught regularly (Bender)
- MMT (Bender, Drellishak, Oepen, and Zabludowski)
  - 10 grammars from the GE class, synced to current Matrix
  - Small domain (17 sentences), using MRS as quasi-interlingua
  - One transfer grammar per language
  - Grammars to be released with LOGON
- Validation (Poulson, Bender, Evans, Drellishak)
  - Customization system allows hundreds of thousands of combos
  - How can we be sure **all** grammars we create parse and generate correctly?
- Morphology (O'Hara)
  - Before, only a few simple prefixes/suffixes
  - Adding support for many kinds of morpheme ordering, optionality, co-occurrence
  - General mechanism for lexical rules, to be used by all future libraries

Upcoming work

- New Libraries
  - Tense and Aspect (Poulson)
  - Case, person, number, and gender on verbs and arguments (Drellishak)
  - Argument optionality, cognitive status of referents, additional word-order variations (Bender)
- Revising coordination
  - Since initial work (Drellishak & Bender 2005), new patterns found
  - Need to account for these in a general way

Issues

- Circular validation?
  - Choose a random language type, make a test suite and test
  - Challenge: the system for creating test suites cannot simply be one of our output grammars used as a generator
  - Our system for making suites is currently based on regular expressions, but must move (at least) to CFGs
- Engineering concerns vs. linguistic analysis
  - Grammars should be linguistically plausible, but also efficient and modular, but these are sometimes in tension
  - E.g. coordination, I have a general-purpose set of rules in mind, but motivated by desire for a single analysis across languages, rather than treating each language individually
- UI for morphology
  - A simple interface for extremely complex interactions
  - Assume we get the slots right, how do we integrate it with paradigms?
  - And what do we call the slots, anyway?