October 28, 2003

Midterm Review

Notes on the exam

- Open book, open notes, closed web
- Covers all material discussed so far

Synthesis

- What is computational linguistics?
- How does it differ from other subfields of linguistics/CS?
- How is it similar to other subfield of linguistics/CS?

Topics covered so far

- Regular expressions
- Finite state automata
- Finite state transducers
- Morphology & morphological parsing
- CFG
- Syntactic parsing
- Feature structures
- Unification
- Parsing with unification

Formal languages

- A formal language is a set of strings
- Things you can do with a formal language:
  - Recognize it
  - Parse it
  - Generate it

Knowledge bases

- Knowledge bases are encodings of (linguistic) information.
- What kinds have we seen in this class?
  - What formal systems to they use?
  - What do they encode?

Using knowledge bases

- Knowledge bases can be used by various algorithms to:
  - recognize
  - parse
  - generate
- ... sets of strings.
- Which algorithms have we seen for each, and what knowledge bases do they use?

Formal devices

- What’s the difference between a deterministic FSA and a non-deterministic FSA?
- What’s the difference between an FSA and an FST?
- In what two contexts have we seen DAGs?

Cascading v. Composing FSTs

- Cascading: output tape of one is input tape of another
- Composing: create one single FST, with one input tape and one output tape
- CF: figs 3.9, 3.10 and 3.11

Four parameters of parsing algorithms

- Top-down v. bottom-up
- Breadth-first v. depth-first
- Best-first v. exhaustive
- Uni- v. bi-directional

Feature structures & unification

- What do we gain from using feature structures?
- What does the operation of unification do?

Topics covered so far

- Regular expressions
- Finite state automata
- Finite state transducers
- Morphology & morphological parsing
- CFG
- Syntactic parsing
- Feature structures
- Unification
- Parsing with unification