October 12, 2004

Chapter 3.3–3.6

Finite State Morphological Parsing

Last time

- Morphology primer
- Using FSAs to recognize morphologically complex words
- FSTs (definition, cascading, composition)
- FSTs for morphological parsing

Overview

- Review of FSTs
- General architecture for morphological processing
- Morphotactics and irregular forms with FSTs
- Spelling change rules
- `xfst`
- Intersection
- Ambiguity
- Lexicon-free morphological analysis
- Human morphological processing

Review: Mealy machines

- Q: a finite set of states $q_0, q_1, \ldots, q_N$
- Σ: a finite alphabet of complex symbols $i : o$ such that $i \in I$ and $o \in O$. $\Sigma \subseteq I \times O$. I and O may each include ϵ.
- q_0: the start state.
- F: the set of final states, $F \subseteq Q$.
- $\delta(q, i : o)$: the transition matrix.

General architecture

- Conceptually three levels of representation:

Lexical	fox +N +Pl
Intermediate	foxˆs
Surface	foxes

- Lexical $\leftrightarrow$ Intermediate: an FST which represents possible concatenations of stems and affixes, and also irregular or suppletive morphemes.
- Intermediate $\leftrightarrow$ Surface: an FST which represents orthographic rewrite rules, run in parallel or composed.
- The whole thing is actually composed into one big FST.

An FST to parse English nouns

- T_{num} (Fig 3.9; p.74) parses the same set of nouns that the FSA in 3.2 recognizes.
- Why does it have more states?
- What are its input and output alphabets?
- The lexicon given for 3.9 has a funny spelling for only two words. Why?

An FST to parse English nouns

- Fig 3.11 (p.76) gives T_{lex}, the result of compiling T_{stems} and T_{num}.
- What sequence of states does T_{lex} go through in parsing the input *goose* and what output does it give?
- What about for *geese*?

A spelling rule FST

- FSTs for orthographic rules model context-sensitive rewrite rules, like (3.5):

$$\epsilon \rightarrow \text{E} \; / \left\{ \begin{array}{c} \text{x} \\ \text{s} \\ \text{z} \end{array} \right\} \hat{} \; \_ \; \text{s\#}$$

- They must change the input only when called for (when their environment is satisfied).
- NB: With rule $\rightarrow$ FST compilers, there's no need to write an FST by hand... (but that doesn't mean there's no need to understand them!)

A spelling rule FST

- Note that their inputs have morpheme and word boundary symbols, while their outputs are standard orthography.
- What states does the FST visit in transducing *foxˆs#* to *foxes*?
- Find other examples that illustrate each of the five states in the machine.
- → `xfst` demo

Building a larger machine

- Figure 3.16 cascades a lexicon FST (T_{lex}, Fig 3.11) with a pile of orthographic rule FSTs (such as $T_{e\text{-}insert}$, Fig 3.14). What does each do?
- How would you use 3.16 to parse a word?
- When would you want to?
- How would you use 3.16 to generate a word?
- When would you want to?
- Does the design allow for orthographic rules which feed each other?

Composition and intersection

- 3.16 cascades one machine that is the result of composing two others, and another machine that is the result of running a whole batch of machines in parallel.
- Intersection allows you to run machines in parallel:
  - Take the Cartesian product of states: $\{q_{ij} \mid q_i \in Q_1,\ q_j \in Q_2\}$
  - For each symbol $a : b$, if that symbol would take machine 1 to q_n and machine 2 to q_m, it takes the combined machine to q_{nm}.
- Play with `xfst` to see why this might be so.

Ambiguity

- Local v. global ambiguity
- Ambiguity in parsing v. generation
- How could you use an FST to give multiple outputs for one input?

What if you don't have a lexicon?

- Why might you not have a (big enough) lexicon?
- Why might you still want to do morphological parsing?
- The Porter stemmer (Appendix B) is a cascade of rewrite rules sensitive to orthographic properties of words, but without knowledge of any particular lexicon.
- Robust systems combine lexicon-based morphological parsing with techniques for handling unknown words. See in particular Morphological Analyzer ChaSen: http://chasen.aist-nara.ac.jp/

Human morphological parsing

- How much morphological analysis do humans do?
- Stanners et al. (1979) and Marslen-Wilson et al. (1994) find evidence for more analysis of inflectional morphology than derivational morphology. How can they tell?
- Speech errors also indicate morphological analysis. How?
- See also Pinker (1999) *Words and Rules*.

Overview

- Review of FSTs
- General architecture for morphological processing
- Morphotactics and irregular forms with FSTs
- Spelling change rules
- `xfst`
- Intersection
- Ambiguity
- Lexicon-free morphological analysis
- Human morphological processing

Coming up...

- Assignment 2 is posted. Look it over.
- Thursday: CFGs and parsing.
- Preliminary choice of final type due Thursday, by email.