Natural Language Processing

Lecture 6

Today's Outline

Administration

- roster
- new announcements and links
- homework

Review and Answers

FSTs continued

Porter Stemmer

Soundex

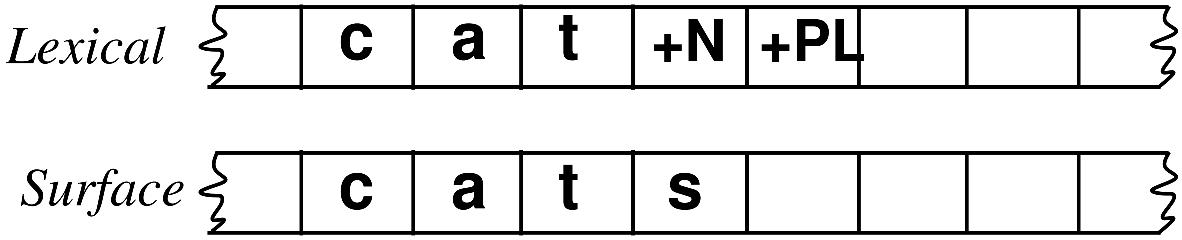
Human Morphology

Review

FSAs for morphological recognition

- lexicon representation

FSTs for morphological parsing



- add an extra tape and extra transition symbols to normal FSAs

Answers from last time

FSTs and Ambiguity (continued)

Unionizable represents a problem involving a global ambiguity.

What's the problem with parsing *assess*?

FSTs: Formal Definition

Like FSAs, FSTs can be formally specified as a 5-tuple:

- a finite set of N states $q_0, q_1 \ldots q_n$: Q
- a finite alphabet of *complex* symbols: Σ (difference from FSA)
  - each complex symbol is composed of an input-output pair i:o
  - the symbol i is from an input alphabet I
  - the symbol o is from an output alphabet 0
- the start state: q_0
- the set of final states: F $\subseteq$ Q
- the transition function that maps Q $\times$ Σ to Q

FSTs (continued)

FSTs are isomorphic to Regular Relations (sets of *pairs* of strings)

Closure properties

- union
- not intersection and others from FSAs
- inversion
  - switches input and output labels
  - converts parser to generator
- composition
  - replaces two transducers that run in series with one that is more complex
  - more later

Some Terminology and Notation

Pairs = complex symbols from alphabet

- a:b

Upper = lexical tape

- left side of a:b pairs

Lower = surface tape

- right side of a:b pairs

Default pairs

- if a=b, that is a:a, write a for shorthand

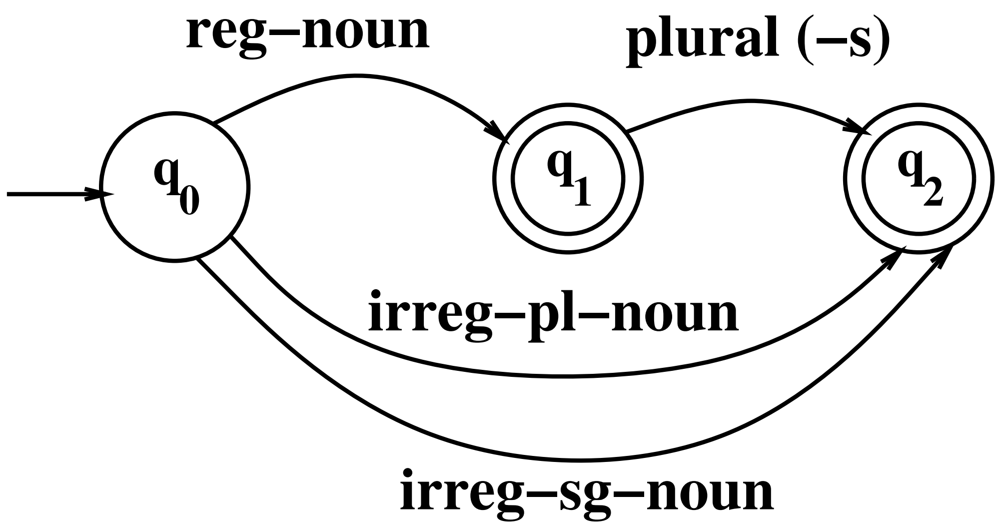
= word boundary

^ = morpheme boundary

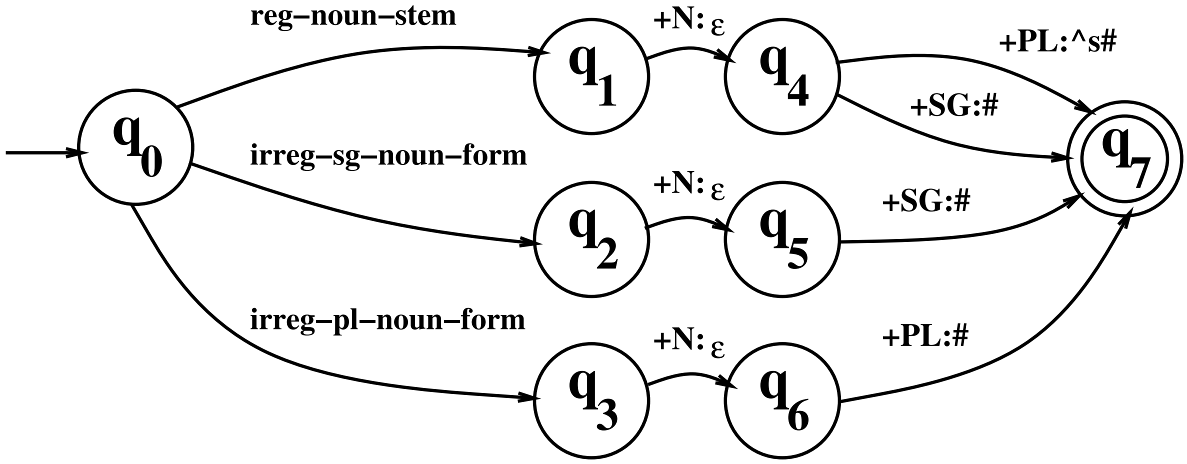
Other = any feasible pair that is not in this transducer

English Nominal Number Inflection

Morphotactic FSA



FST: Add lexical tape and morphological features



Embed a lexicon, using two levels (irregular plurals)

- g:g o:e o:e s:s e:e (equivalently g o:e o:e s e)

Multi-Level Multi-Tape Machines

A frequently used FST idiom, called a Cascade, is to have the output of one FST read in as the input to a subsequent machine.

Thus, for irregular plural nouns, cascade the lexicon FST with the nominal number inflection FST.

Spelling

Spelling is a problem we have been glossing over.

- `cat+N+PL` $\rightarrow$ `cats` is ok
- `fox+N+PL` $\rightarrow$ `foxs` is not

English often requires spelling changes at morpheme boundaries.

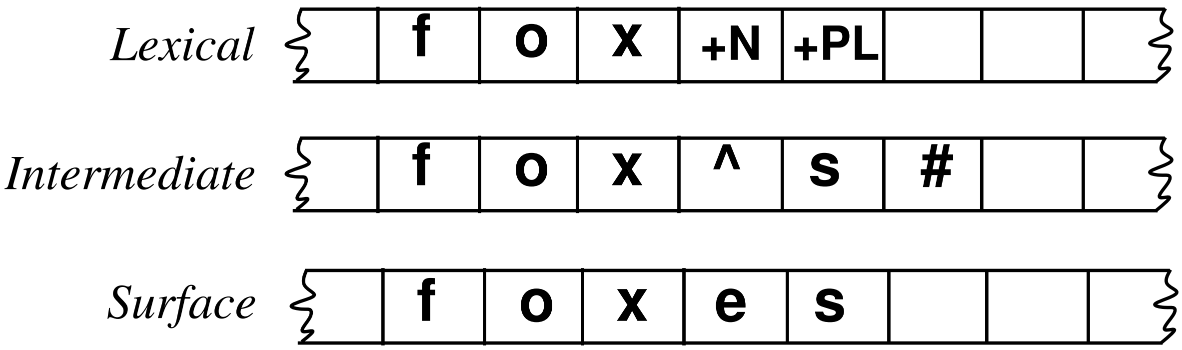
Thus, context-specific <u>spelling rules</u> (or <u>orthographic rules</u>) have to be handled.

Spelling Rules

See table on page 77 of Jurafsky and Martin

Multi-Level Multi-Tapes (cont.)

So to handle spelling we use three tapes:

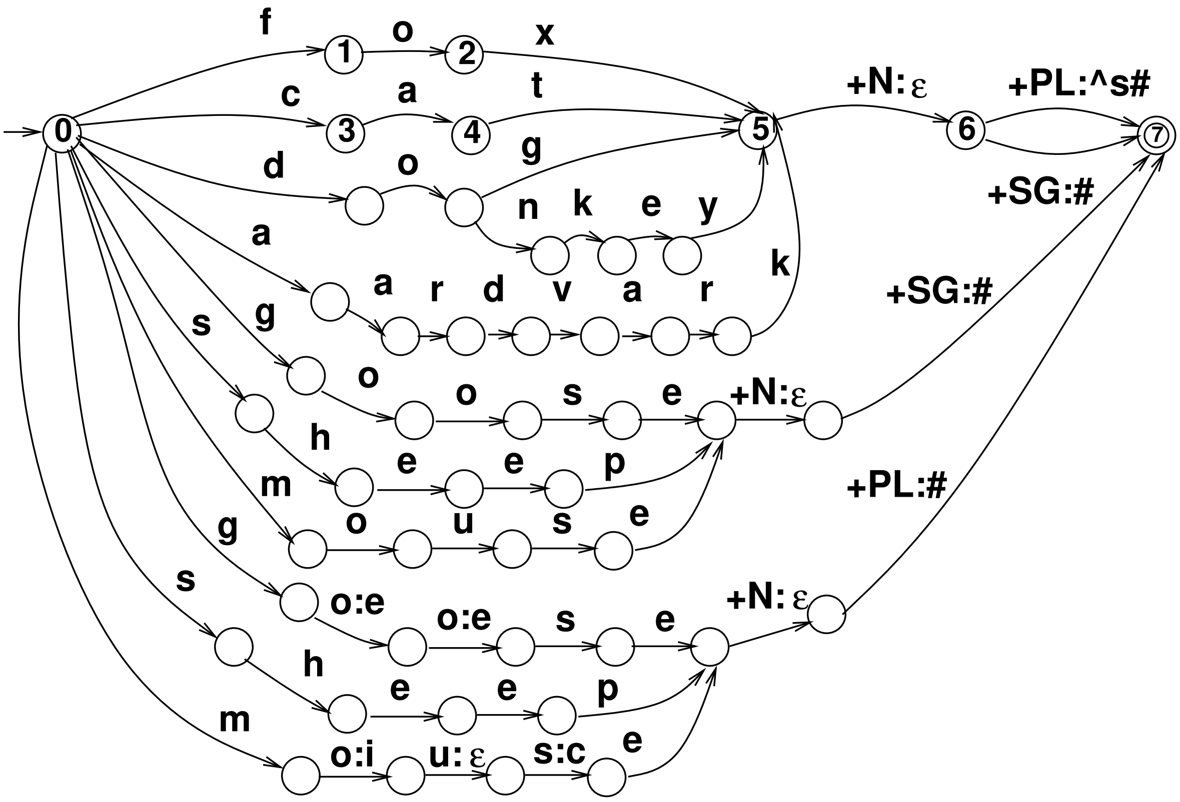


We need one transducer to work between the lexical and intermediate levels and a second (bunch) to work between the intermediate and surface levels to patch up the spelling (e.g., to do e-insertion).

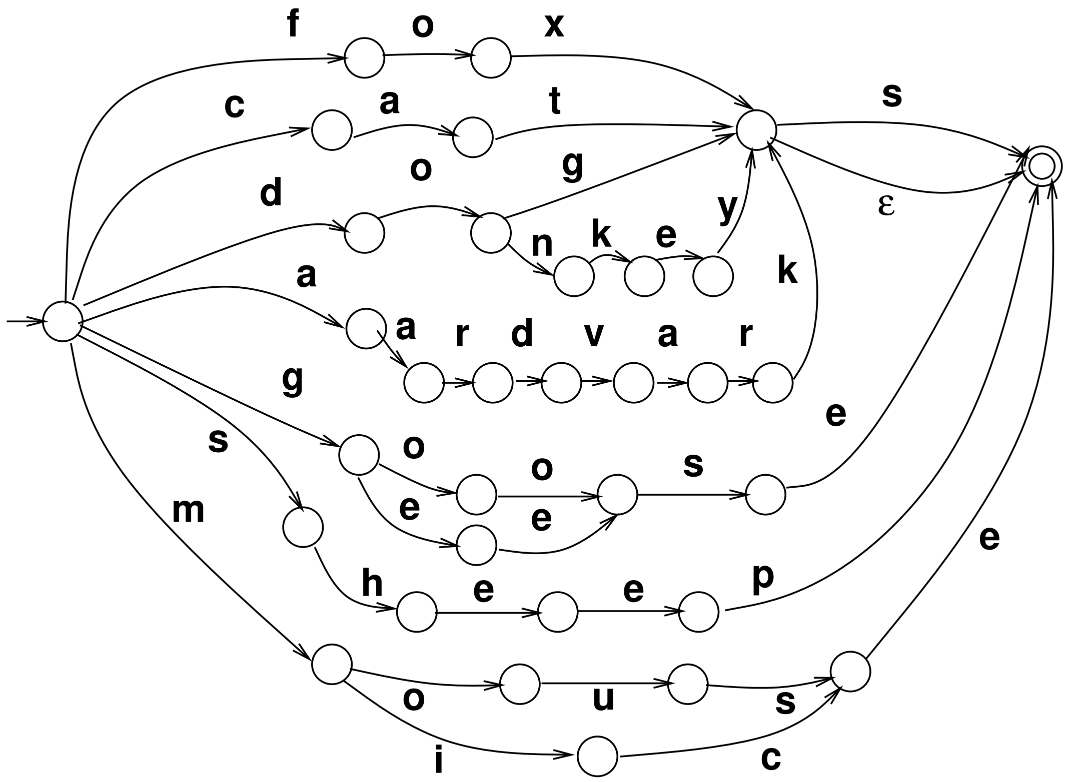
Intermediate tapes: tapes with boundary markers

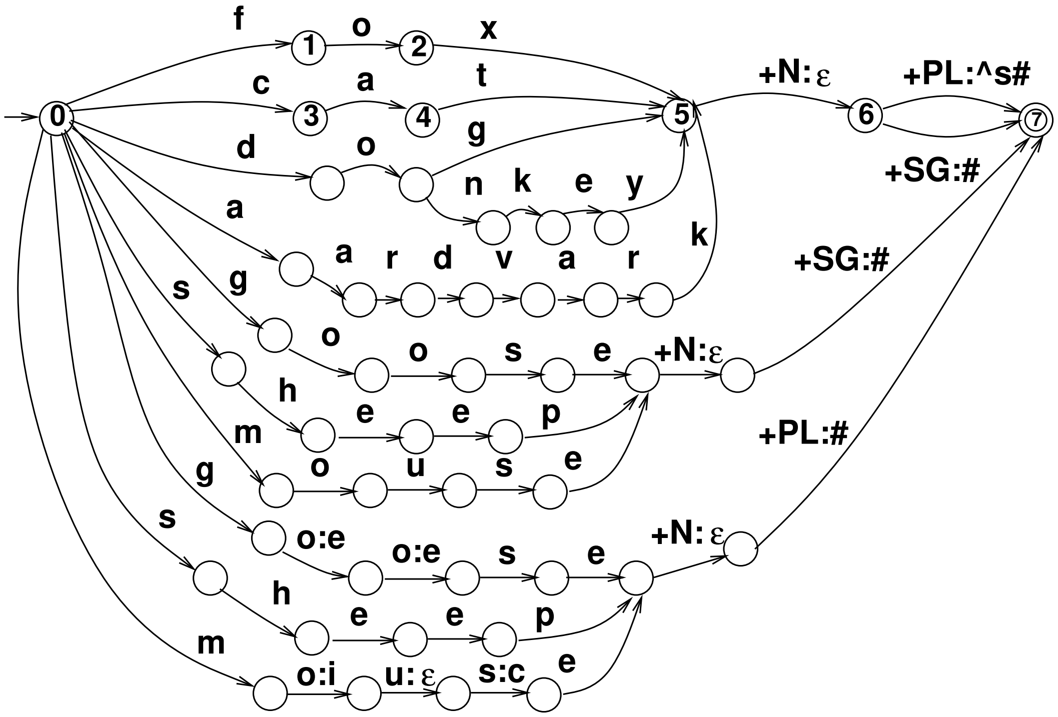
Lexical to Intermediate Transducer

After composition...



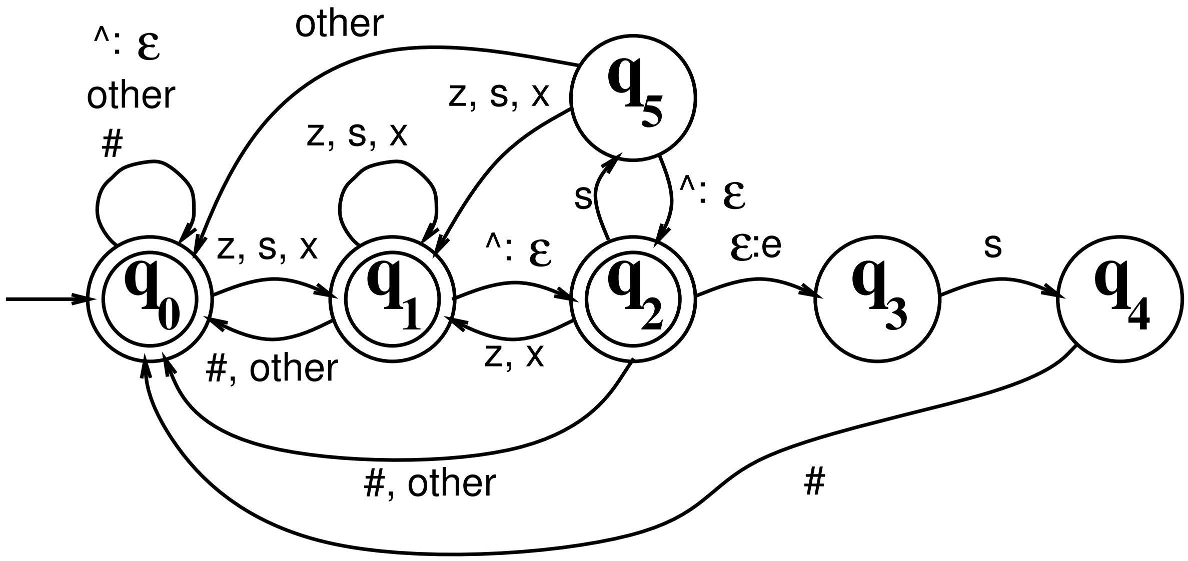
Nominal Inflection (FSA vs. FST)





Intermediate to Surface Transducer

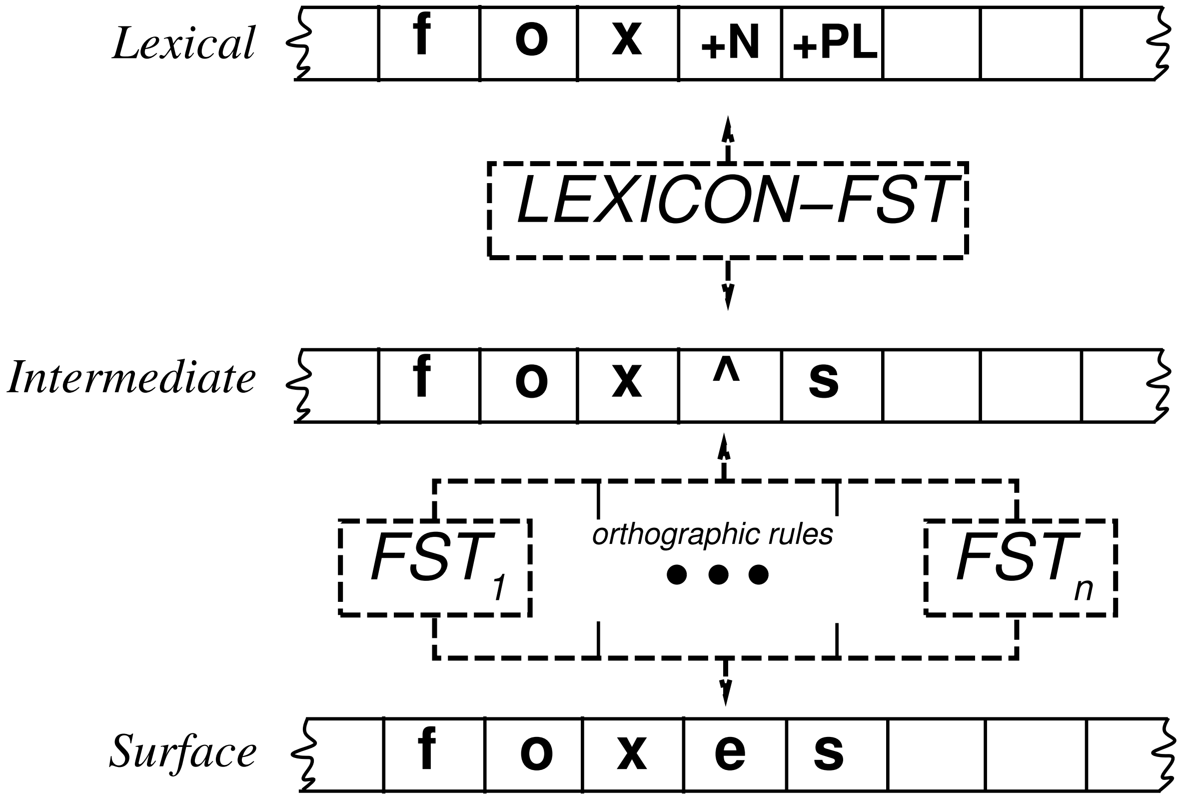
We need an FST to add an e between an x, s or z and before the s at morpheme boundary at the end of a word.



A key point of this transducer is that irrelevant stuff passes through unchanged.

Combining FST Lexicon and Rules

The final result is the following theoretical machine:



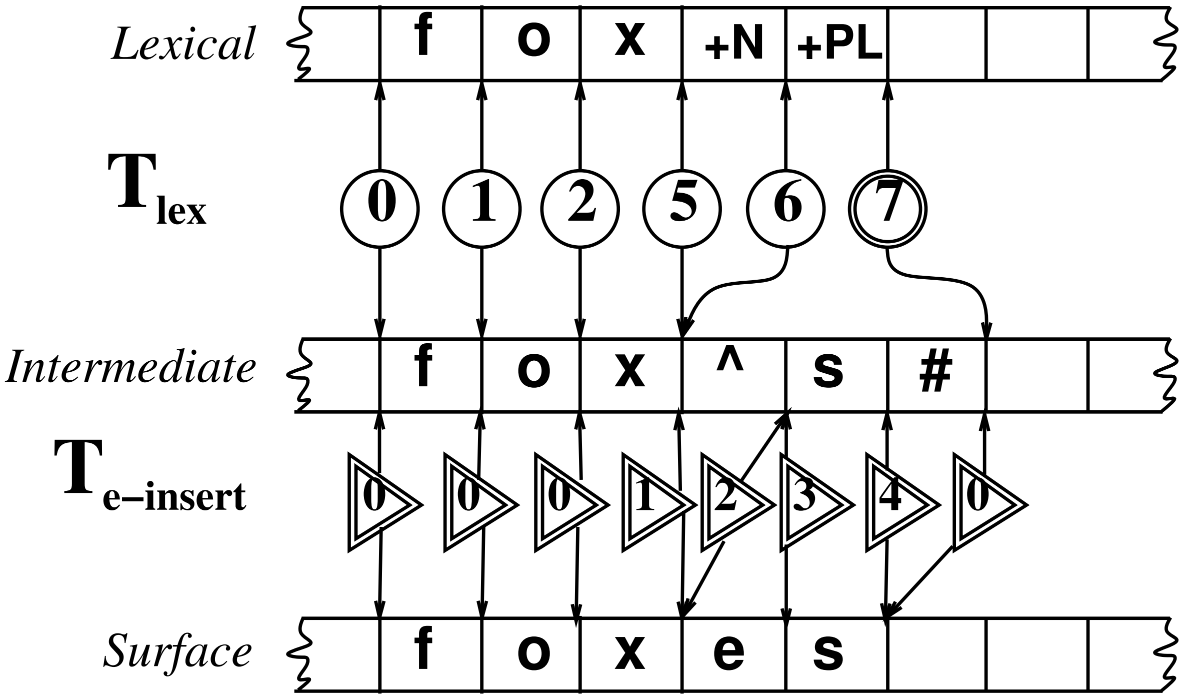
A 2 level cascaded architecture

- generation: use top-down
- parsing: use bottom-up

Accepting Foxes

The circles are states in the lexical to intermediate machine.

The triangles are states in the intermediate to surface machine.



Composition

Actually running a cascade as a cascade can turn out to be a pain...

- it is hard to manage all the tapes
- it fails to take advantage of the restricting power of all the machines

So...

- it is better to compile the cascade into a single large machine with two tapes (input and output)

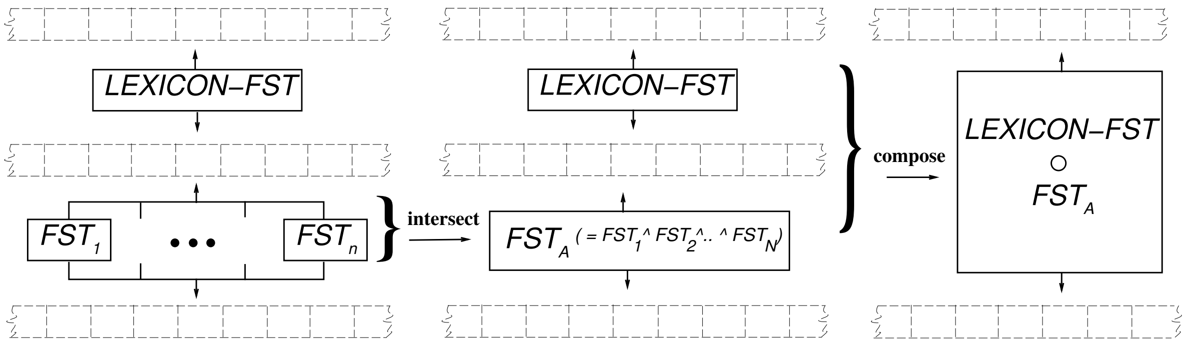
Composition Construction

Create a new state (x, y) for every pair of states $x \in Q_1$ and $y \in Q_2$. And give this new machine a new transition function (as detailed in the book).

The intermediate tapes go away, as do the intermediate output symbols.

Intersection and Composition

Putting everything together...



Compose a cascade of transducers in series into a single more complex transducer

Transducers in parallel can be combined by intersection.

Information Retrieval

As we have discussed, information retrieval (IR) is an application that has made use of morphological processing.

In IR the key is to acquire the stems, not to make any real use of morphological structure, hence the term Stemming.

Small performance improvements in practice

- stemming introduces both kinds of errors (e.g., organization $\rightarrow$ organ)
- with larger documents, less needed

Porter Stemming: Lexicon-free FST

A problem is that you often don't have a true lexicon...

- ad hoc morphology without access to a lexicon, in the form of *cascaded rewrite rules*
- IZER $\rightarrow$ IZE
- digitizer $\rightarrow$ digitize

See Appendix B in Jurafsky and Martin for details.

Soundex

The Ellis Island problem. It's extremely hard to uniformly transcribe rare words (i.e., names). Hence the Soundex method (see Jurafsky and Martin page 89, 3.6), which can be implemented as a FST.

1. Keep the first letter of the name
2. Drop non-initial a,e,h,i,o,u,w,y
3. Replace remaining letters
   - b,f,p,v $\rightarrow$ 1
   - c,g,j,k,q,s,x,z $\rightarrow$ 2
   - ...
   - r $\rightarrow$ 6
4. Replace sequences of identical numbers with a single number
5. ...

Thus, Jurafsky, Jaarofsky, Jarovsky, and Jarovski all map to J612.

Human Morphological Processing

Are lexicons represented using Full Listing or Minimal Redundancy (as in this chapter)?

Evidence suggests both representations are used.

- priming
- slips of the tongue ("easy enoughly" instead of "easily enough")

For Next Time

Chapter 6

Acknowledgements

Some of this material was taken from the course web pages of James Martin.