CS674 Natural Language Processing

- Last class
  - Need for morphological analysis
  - Basics of English morphology
  - Finite-state morphological parsing
    - Introduction

Goal

- Input: surface form
- Output: stem plus morphological features
- Focus: productive nominal plural (-*s*)
  verbal progressive (-*ing*)
  - foxes → fox +N +PL
  - geese → goose +N +PL
  - eating → eat +V +PRES-PART
  - goose → (goose +N +SG) or (goose +V)

What knowledge sources will we need?

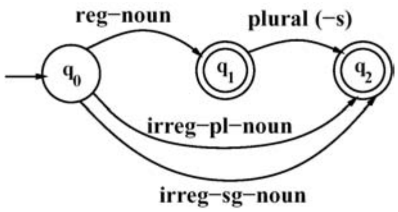
- Lexicon
  - List of stems and affixes with basic information about each
- Morphotactics
  - Model of morpheme ordering
  - Explains which classes of morphemes can follow others
- Spelling rules
  - Orthographic rules
  - Model the spelling changes that occur in a word when two morphemes combine

Topics for today

- Finite-state morphological parsing
  - **Lexicon and morphotactics**
  - Morphological parsing with FST's
  - Orthgraphic rules
  - Combining it all

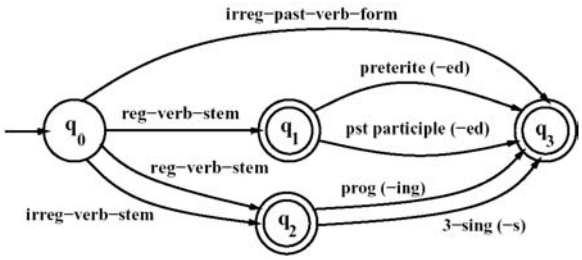
The lexicon

- Usually not represented as a list of words
- Structured as
  - List of stems and affixes
  - Representation of the morphotactics
- Represent via a finite-state automaton (J&M Ch. 2)



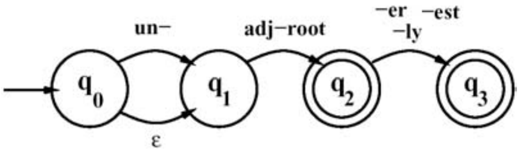
J&M Fig 3.2

Verbal inflection



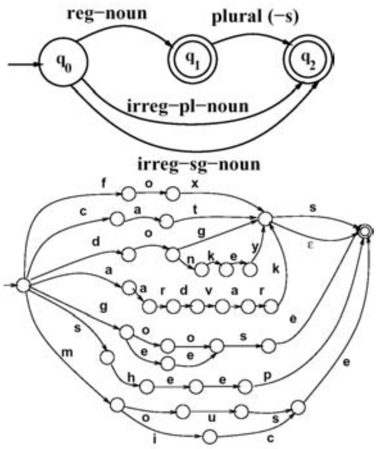
FSA's for derivational morphology

- Much more complex
- Often use CFG's instead
- Consider adjective morphology…what's the problem?



FSA's for morphological recognition

- Goal: Use the FSA's to determine whether an input string of letters makes up a legitimate English word
  - Combine the list of stems with the FSA
  - Expand each arc with all of the morphemes that comprise the class



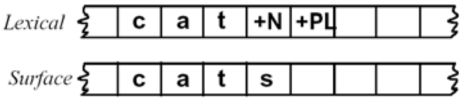
Topics for today

- Finite-state morphological parsing
  - Lexicon and morphotactics
  - Morphological parsing with FST's
  - Orthgraphic rules
  - Combining it all

Two-level morphology

- Represents a word as a correspondence between
  - Surface level
    - **Represents the spelling of the word, i.e. letter sequences**
  - Lexical level
    - **Represents a concatenation of morphemes, i.e. morpheme and feature sequences**

Two-level morphology example



Lexical	c	a	t	+N	+PL			
Surface	c	a	t	s				

- **Mapping between the two levels is accomplished via a finite-state transducer (FST)**

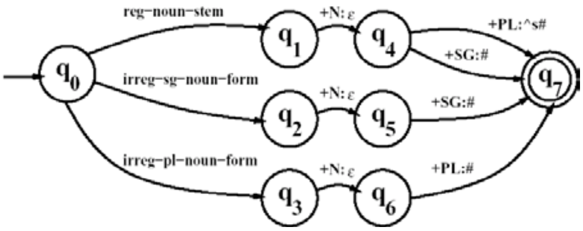
Finite-state transducers

- A finite-state automaton that maps between one set of symbols and another
- An FSA defines a formal language by defining a set of strings
- Defines a *relation* between sets of strings
- Reads one string and generates another

Formal definition

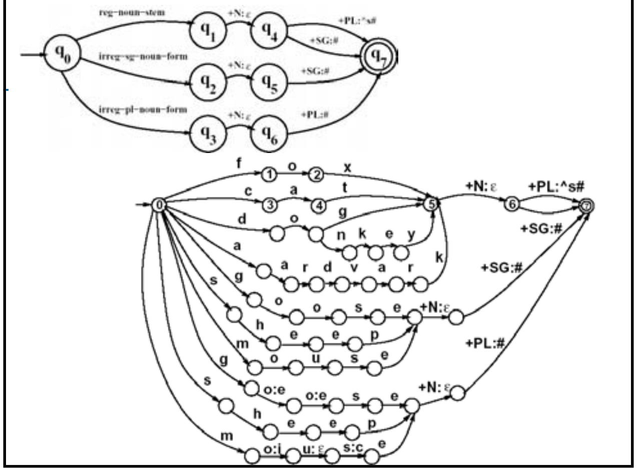
- *Q*: a finite set of *N* states $q_0, q_{1,\ldots,} q_N$
- q_0: start state
- *F*: set of final states
- $\sum$: a finite alphabet of input-output pairs *i:o*
- δ*(q,i:o)*: transition function between states. Given a state $q \in Q$and complex symbol *i:o,* δ*(q,i:o)* returns a new state $q' \in Q$

FST morphological parser

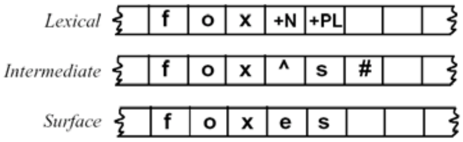


Two-level lexicon

- reg-noun
  - tree
  - cloud
- irreg-pl-noun
  - g o:e o:e s e
  - sheep
  - m o:l u:ε s:c e
- irreg-sg-noun
  - goose
  - sheep
  - mouse



Lexical and intermediate tapes

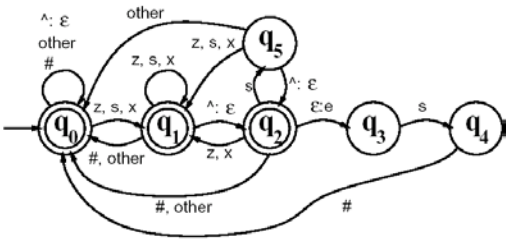


Lexical	f	o	x	+N	+PL			
Intermediate	f	o	x	^	s	#		
Surface	f	o	x	e	s			

Orthographic Rules

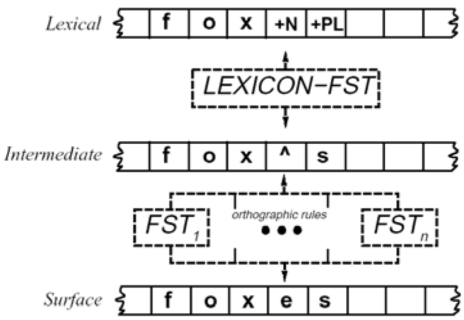
- E insertion (for example)
  - e added after *–s, -z, -x, -ch, -sh* before *–s*
    - watch/watches
    - fox/foxes
- Implement these rules as a *cascade* of FST's
  - Output of one transducer is the input to the next transducer
  - One transducer per orthographic rule
  - Each transducer needs to express the constraints necessary for that rule; allow any other string of symbols to pass through unchanged.

Transducer for E-insertion



Topics for today

- Finite-state morphological parsing
  - Lexicon and morphotactics
  - Morphological parsing with FST's
  - Orthgraphic rules
  - Combining it all



Ambiguity

- *foxes* can be a verb as well as a noun
- Local ambiguities occur
  - E.g. *caress*
- What shall we do?
  - Non-determinism requires the FST-parsing algorithm to include a search algorithm