Morphology

Jugal Kalita, University of Colorado at Colorado Springs

Adapted from Kathy McCoy, University of Delaware

September 2010

What is Morphology?

- The study of how words are composed of morphemes (the smallest meaning-bearing units of a language)
  - Stems – core meaning units in a lexicon
  - Affixes (prefixes, suffixes, infixes, circumfixes) – bits and pieces that combine with stems to modify their meanings and grammatical functions (can have multiple ones)
    - Immaterial
    - Trying
    - Absobl**dylutely, Man-f**king-hattan (infixing bloody or fucking)
    - Unreadable

Why is Morphology Important to the Lexicon?

Full listing versus Minimal Redundancy

- true, truer, truest, truly, untrue, truth, truthful, truthfully, untruthfully, untruthfulness
- Untruthfulness = un- + true + -th + -ful + -ness
- These morphemes appear to be productive
- By representing knowledge about the internal structure of words and the rules of word formation, we can save room and search time.

Need to do Morphological Parsing

Morphological Parsing (or Stemming)

- Taking a surface input and breaking it down into its morphemes
- foxes breaks down into the morphemes fox (noun stem) and –es (plural suffix)
- rewrites breaks down into re- (prefix) and write (stem) and –s (suffix)

Two Broad Classes of Morphology

- Inflectional Morphology
  - Combination of stem and morpheme resulting in word of same class
  - Usually fills a syntactic feature such as agreement
  - E.g., plural –s, past tense -ed
- Derivational Morphology
  - Combination of stem and morpheme usually results in a word of a different class
  - Meaning of the new word may be hard to predict
  - E.g., +ation in words such as computerization

Word Classes

- By word class, we have in mind familiar notions like noun and verb that we discussed a bit in the previous lecture.
- Right now we're concerned with word classes because the way that stems and affixes combine is based to a large degree on the word class of the stem.

English Inflectional Morphology

- Word stem combines with grammatical morpheme
  - Usually produces word of same class
  - Usually serves a syntactic function (e.g., agreement)

    like → likes or liked

    bird → birds
- Nominal morphology
  - Plural forms
    - s or es
    - Irregular forms (next slide)
    - Mass vs. count nouns (email or emails)
  - Possessives

Complication in Morphology

- It can get a little complicated by the fact that some words misbehave (refuse to follow the rules)
- The terms regular and irregular will be used to refer to words that follow the rules and those that don't.

Regular (Nouns)

- Singular (cat, thrush)
- Plural (cats, thrushes)
- Possessive (cat's thrushes')

Irregular (Nouns)

- Singular (mouse, ox)
- Plural (mice, oxen)

- Verbal inflection
  - *Main* verbs (sleep, like, fear) are relatively regular
    - -s, ing, ed
    - And productive (i.e., can be used with newly formed verbs): Emailed, instant-messaged, faxed, Imed, SMSed
    - But eat/ate/eaten, catch/caught/caught: These are irregular.
  - *Primary* (be, have, do) and *modal* verbs (can, will, must) are often irregular and not productive
    - Be: am/is/are/were/was/will/been/being
  - Irregular verbs few (~250) but frequently occurring
  - English verbal inflection is much simpler than e.g., Latin, Sanskrit or German

Regular and Irregular Verbs

- Regulars…
  - Walk, walks, walking, walked, walked
- Irregulars
  - Eat, eats, eating, ate, eaten
  - Catch, catches, catching, caught, caught
  - Cut, cuts, cutting, cut, cut

Derivational Morphology

- Derivational morphology is somewhat messy.
  - There is usually only a partial pattern to what is acceptable
  - Irregular meaning change
  - Changes of word class

English Derivational Morphology

- Word stem combines with grammatical morpheme
  - Usually produces a word of a **different** class
  - More complicated than inflectional
- Example: nominalization
  - -ize verbs → -ation nouns
  - generalize, realize → generalization, realization
  - verb → -er nouns
  - Murder, spell → murderer, speller
- Example: verbs, nouns → adjectives
  - embrace, pity→ embraceable, pitiable
  - care, wit → careless, witless

- Example: adjective → adverb
  - happy → happily
- More complicated to model than inflection
  - Less productive: *science-less, *concern-less, *go-able, *sleep-able
  - It's difficult to know what works with which types of words
  - Meanings of derived terms harder to predict by rule
    - clueless, careless, nerveless

Derivational Examples

- Verb/Adj to Noun

-ation	computerize	computerization
-ee	appoint	appointee
-er	kill	killer
-ness	fuzzy	fuzziness

Derivational Examples

- Noun/Verb to Adj

-al	Computation	Computational
-able	Embrace	Embraceable
-less	Clue	Clueless

Compute

- Many paths are possible…
- Start with compute
  - Computer -> computerize -> computerization
  - Computation -> computational
  - Computer -> computerize -> computerizable
  - Compute -> computee

Parsing

- Taking a surface input and identifying its components and underlying structure
- Morphological parsing: parsing a word into stem and affixes and identifying the parts and their relationships
  – Stem and **features**:
    - goose → goose +N +SG or goose +V
    - geese → goose +N +PL
    - gooses → goose +V +3SG
  – Bracketing: indecipherable → [in [[de [cipher]] able]]

Why parse words?

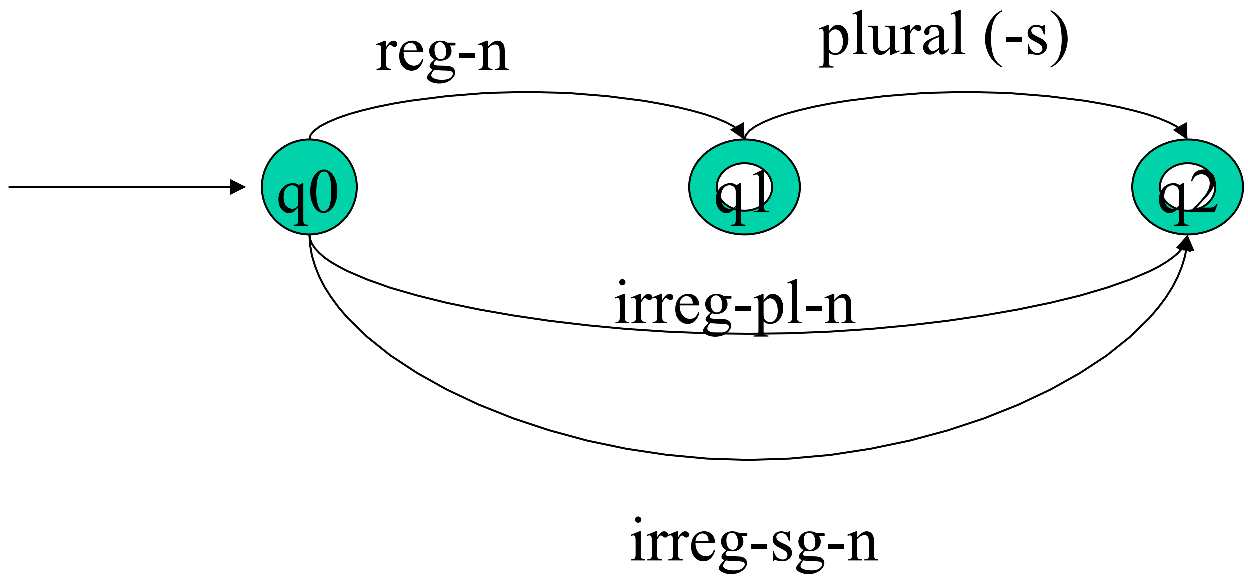
- For spell-checking
  - Is muncheble a legal word?
- To identify a word's part-of-speech (pos)
  - For **sentence parsing**, for **machine translation**, …
- To identify a word's stem
  - For **information retrieval**

What do we need to build a morphological parser?

- Lexicon: stems and affixes (w/ corresponding pos)
- Morphotactics of the language: model of the order in which morphemes can be affixed to a stem. E.g., plural morpheme follows noun in English
- Orthographic rules: spelling modifications that occur when affixation occurs
  - in → il in context of l (in- + legal)

Morphotactic Models

- English nominal inflection



•Inputs: cats, goose, geese

Antworth data on English Adjectives

- Big, bigger, biggest
- Cool, cooler, coolest, cooly
- Red, redder, reddest
- Clear, clearer, clearest, clearly, unclear, unclearly
- Happy, happier, happiest, happily
- Unhappy, unhappier, unhappiest, unhappily
- Real, unreal, really

Antworth data on English Adjectives

- Big, bigger, biggest
- Cool, cooler, coolest, cooly
- Red, redder, reddest
- Clear, clearer, clearest, clearly, unclear, unclearly
- Happy, happier, happiest, happily
- Unhappy, unhappier, unhappiest, unhappily
- Real, unreal, really

Antworth data on English Adjectives

- Big, bigger, biggest
- Cool, cooler, coolest, cooly
- Red, redder, reddest
- Clear, clearer, clearest, clearly, unclear, unclearly
- Happy, happier, happiest, happily
- Unhappy, unhappier, unhappiest, unhappily
- Real, unreal, really

Antworth data on English Adjectives

- Big, bigger, biggest
- Cool, cooler, coolest, cooly
- Red, redder, reddest
- Clear, clearer, clearest, clearly, unclear, unclearly
- Happy, happier, happiest, happily
- Unhappy, unhappier, unhappiest, unhappily
- Real, unreal, really

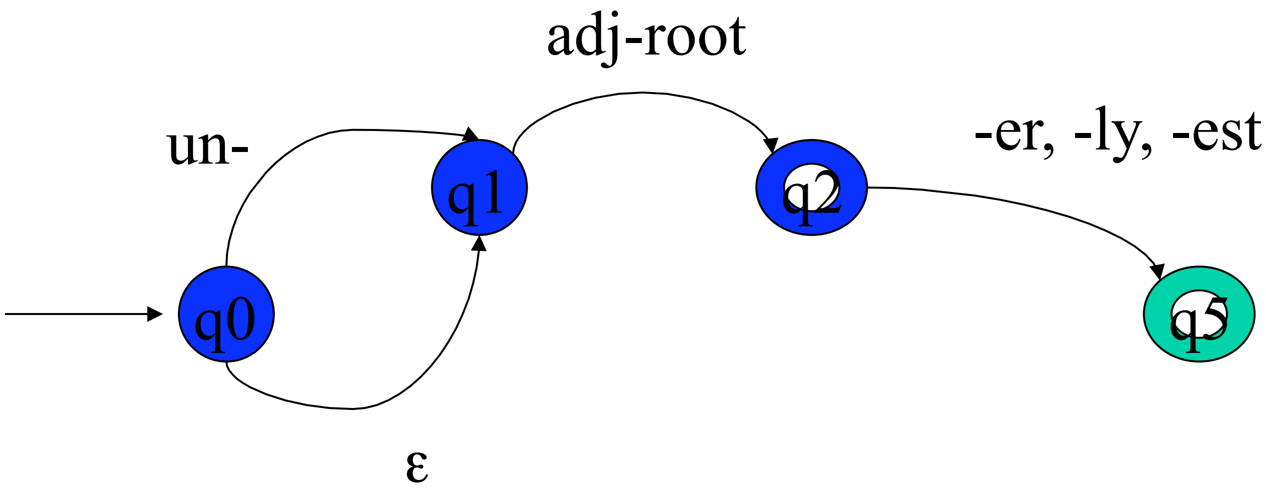
Antworth data on English Adjectives

- Big, bigger, biggest
- Cool, cooler, coolest, cooly
- Red, redder, reddest
- Clear, clearer, clearest, clearly, unclear, unclearly
- Happy, happier, happiest, happily
- Unhappy, unhappier, unhappiest, unhappily
- Real, unreal, really

Antworth data on English Adjectives

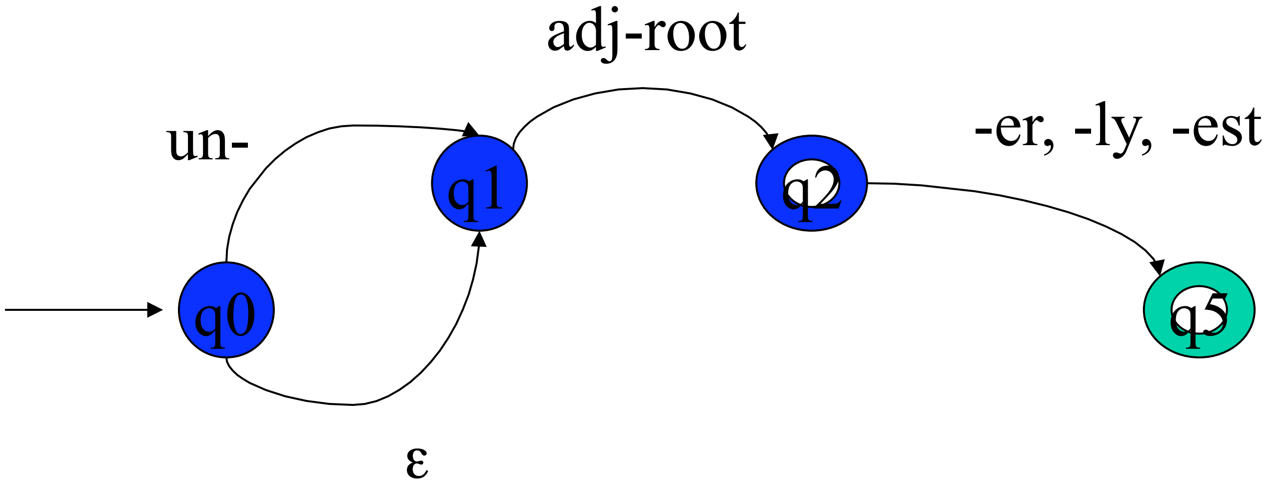
- Big, bigger, biggest
- Cool, cooler, coolest, cooly
- Red, redder, reddest
- Clear, clearer, clearest, clearly, unclear, unclearly
- Happy, happier, happiest, happily
- Unhappy, unhappier, unhappiest, unhappily
- Real, unreal, really

- Derivational morphology: adjective fragment



- Adj-root: clear, happy, real, big, red

- Derivational morphology: adjective fragment



- Adj-root: clear, happy, real, big, red

•BUT: unbig, redly, realest

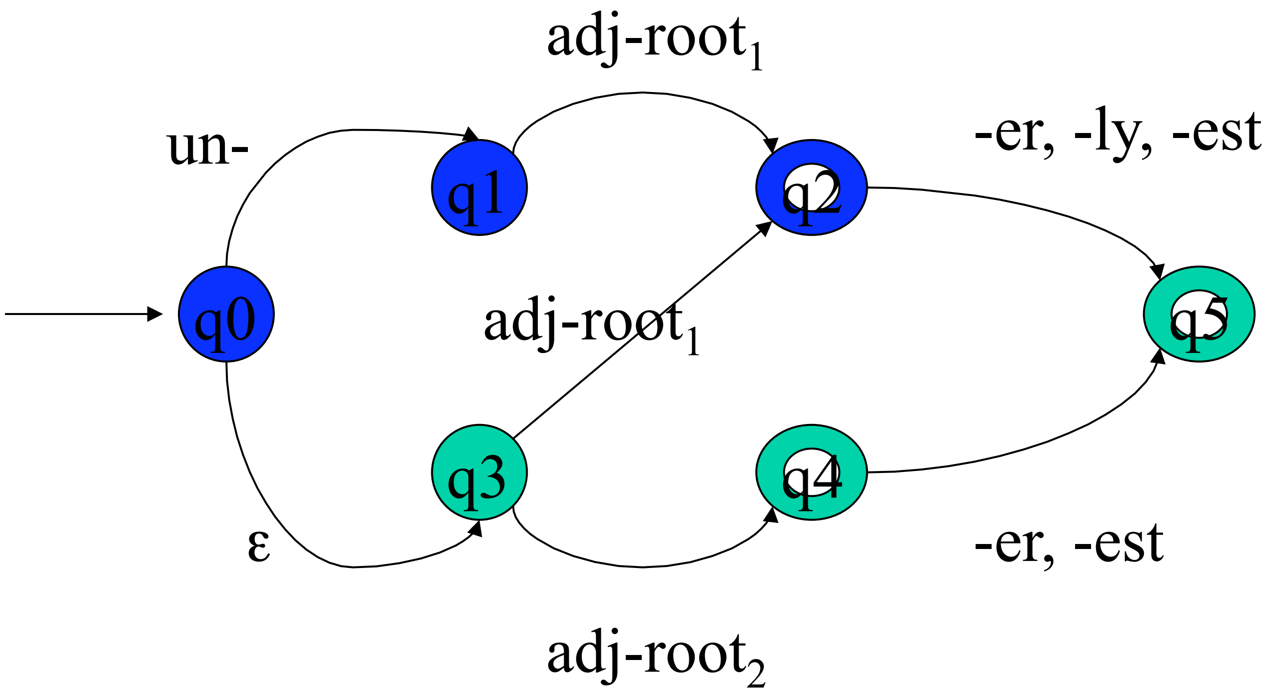
Antworth data on English Adjectives

- Big, bigger, biggest
- Cool, cooler, coolest, cooly
- Red, redder, reddest
- Clear, clearer, clearest, clearly, unclear, unclearly
- Happy, happier, happiest, happily
- Unhappy, unhappier, unhappiest, unhappily
- Real, unreal, really

Antworth data on English Adjectives

- Big, bigger, biggest
- Cool, cooler, coolest, cooly
- Red, redder, reddest
- Clear, clearer, clearest, clearly, unclear, unclearly
- Happy, happier, happiest, happily
- Unhappy, unhappier, unhappiest, unhappily
- Real, unreal, really

- Derivational morphology: adjective fragment



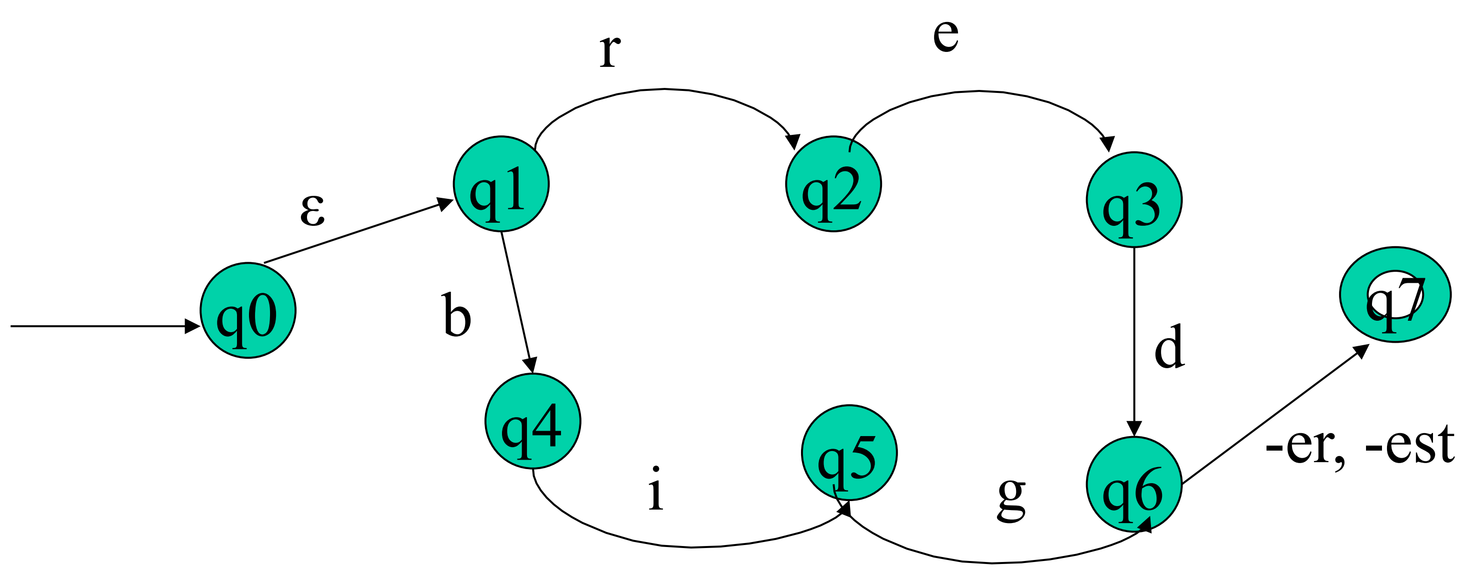
- Adj-root$_1$: clear, happy, real
- Adj-root$_2$: big, red

FSAs and the Lexicon

- First we'll capture the morphotactics
  - The rules governing the ordering of affixes in a language.
- Then we'll add in the actual words

Using FSAs to Represent the Lexicon and Do Morphological Recognition

- Lexicon: We can expand each non-terminal in our NFSA into each stem in its class (e.g. adj_root_2 = {big, red}) and expand each such stem to the letters it includes (e.g. red → r e d, big → b i g)



Limitations

- To cover all of e.g. English will require very large FSAs with consequent search problems
  - Adding new items to the lexicon means recomputing the FSA
  - Non-determinism
- FSAs can only tell us whether a word is in the language or not – what if we want to know more?
  - What is the stem?
  - What are the affixes and what sort are they?
  - We used this information to build our FSA: can we get it back?

Parsing/Generation vs. Recognition

- Recognition is usually not quite what we need.
  - Usually if we find some string in the language we need to find the structure in it (parsing)
  - Or we have some structure and we want to produce a surface form (production/generation)
- Example
  - From "`cats`" to "`cat +N +PL`"

Finite State Transducers

- The simple story
  - Add another tape
  - Add extra symbols to the transitions

  - On one tape we read "`cats`", on the other we write "`cat +N +PL`"

Parsing with Finite State Transducers

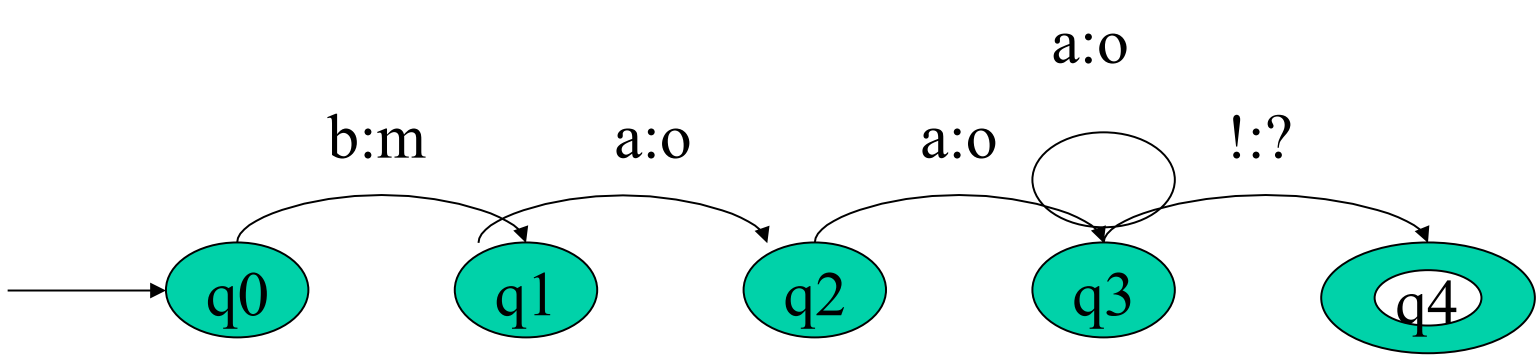
- cats → cat +N +PL
- Kimmo Koskenniemi's two-level morphology
  - Words represented as correspondences between **lexical** level (the morphemes) and **surface** level (the orthographic word)
  - Morphological parsing: building **mappings** between the lexical and surface levels

	c	a	t	+N	+PL	
	c	a	t	s		

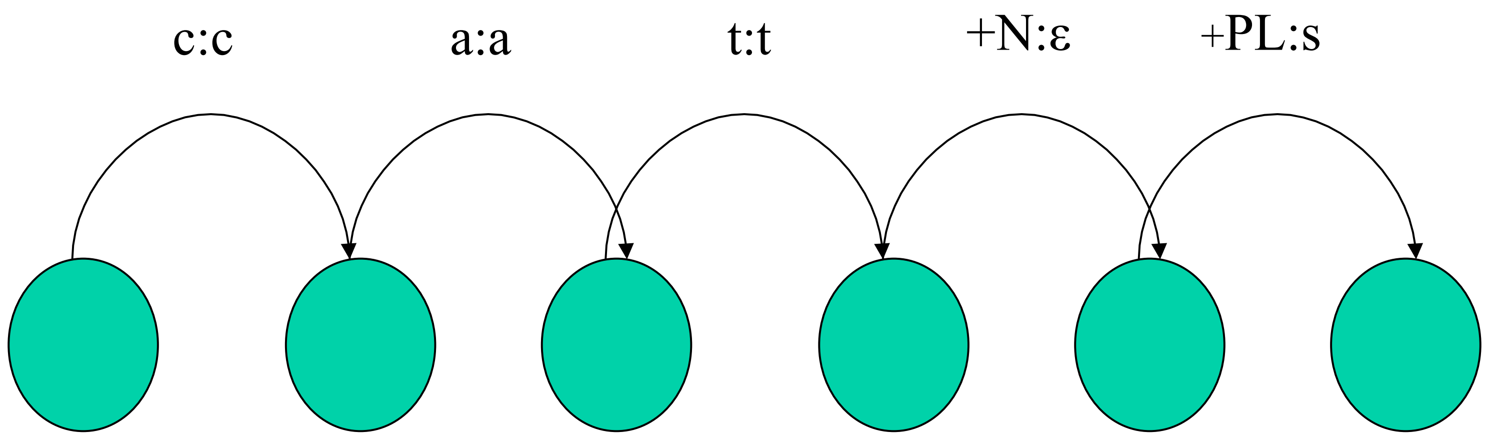
Finite State Transducers

- FSTs map between one set of symbols and another using an FSA whose alphabet Σ is composed of pairs of symbols from input and output alphabets
- In general, FSTs can be used for
  - Translator (Hello:Ciao)
  - Parser/generator (Hello:How may I help you?)
  - To map between the lexical and surface levels of Kimmo's 2-level morphology

- FST is a 5-tuple consisting of
  - Q: set of states {q0,q1,q2,q3,q4}
  - Σ: an alphabet of complex symbols, each an i/o pair s.t. $i \in I$ (an input alphabet) and $o \in O$ (an output alphabet) and Σ is in I x O
  - q0: a start state
  - F: a set of final states in Q {q4}
  - δ(q,i:o): a transition function mapping Q x Σ to Q
  - **Emphatic Sheep → Quizzical Cow**



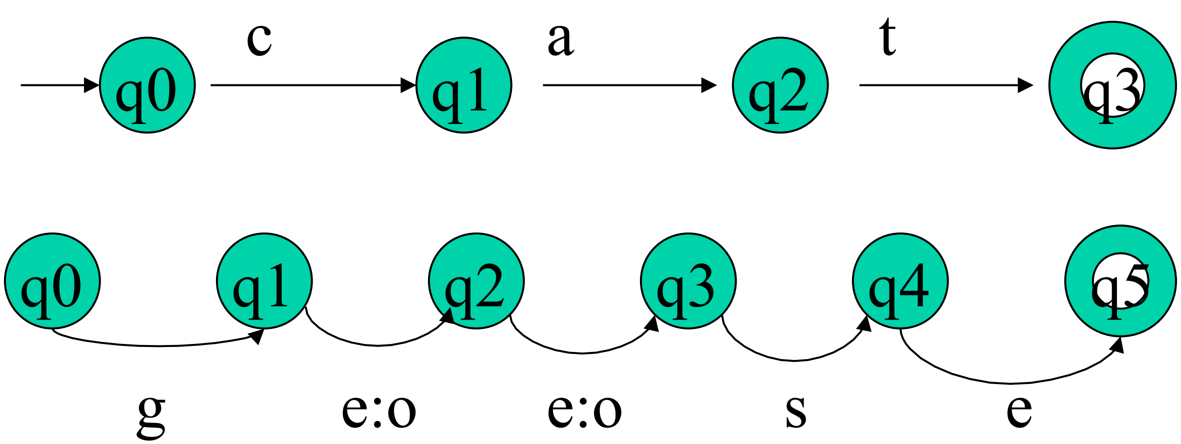
Transitions



- c:c means read a c on one tape and write a c on the other
- +N:ε means read a +N symbol on one tape and write nothing on the other
- +PL:s means read +PL and write an s

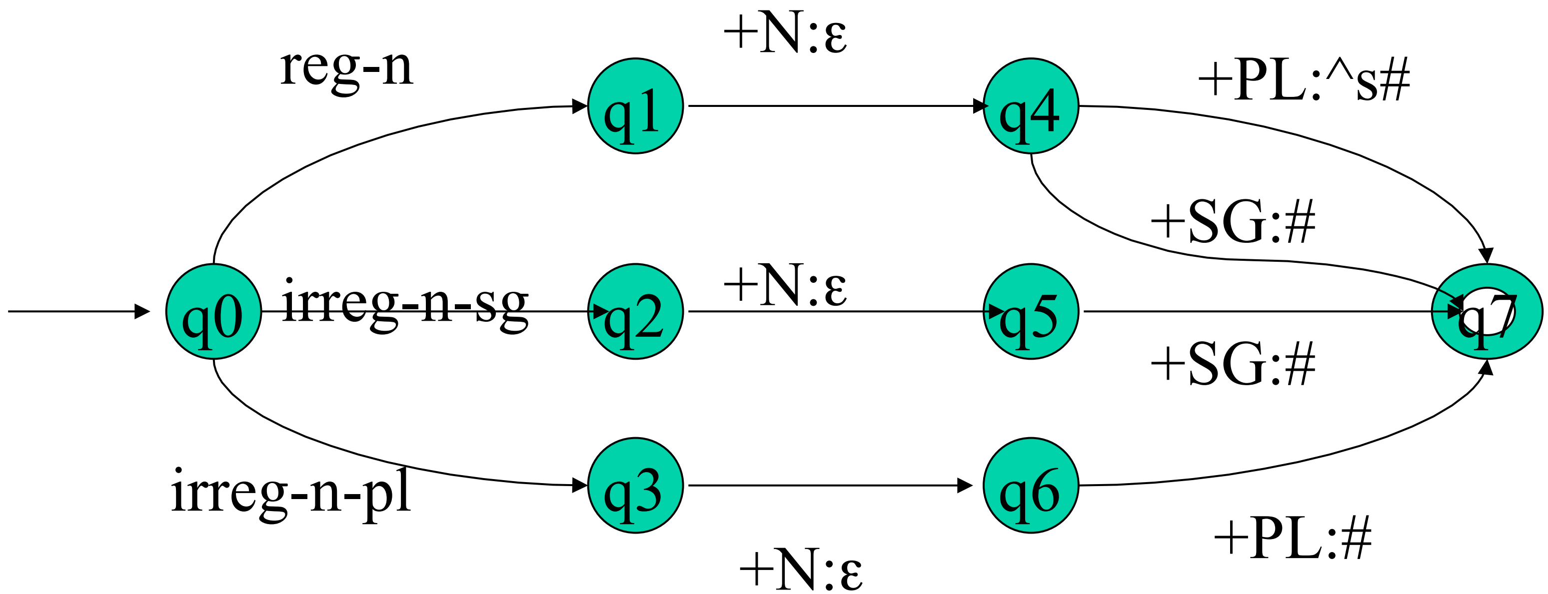
FST for a 2-level Lexicon

- E.g.



Reg-n	Irreg-pl-n	Irreg-sg-n
c a t	g o:e o:e s e	g o o s e

FST for English Nominal Inflection



Combining (cascade or composition) this FSA with FSAs for each noun type replaces e.g. reg-n with every regular noun representation in the lexicon. ^ is morpheme boundary; # is word boundary

Problems with a 1-level FST

- Of course, its not as easy as
  - `“cat +N +PL” <-> “cats”`
- Or even dealing with the irregulars geese, mice and oxen
- But there are also a whole host of spelling/ pronunciation changes that go along with inflectional changes

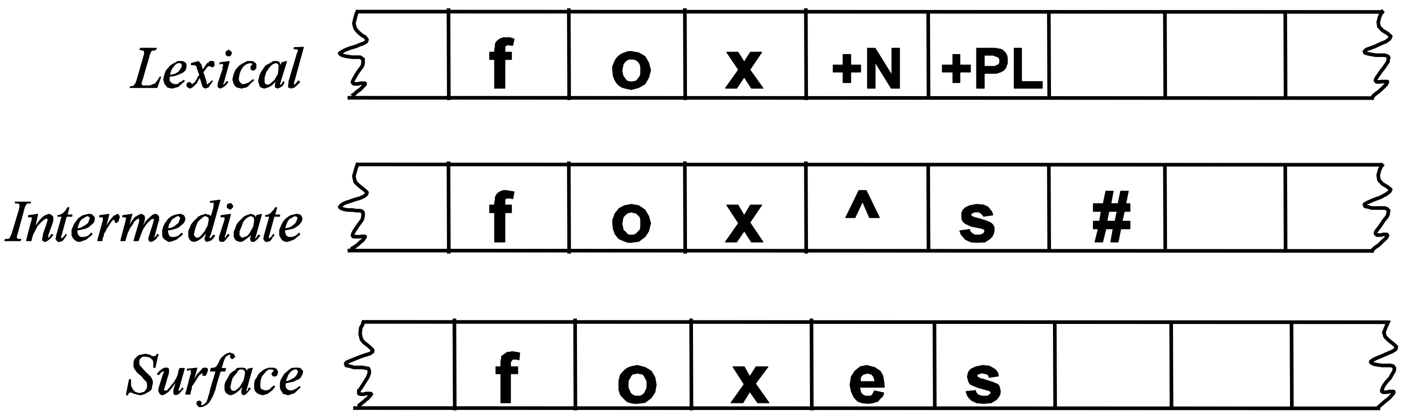
Examples of Spelling Changes

Name	Rule	Example
Consonant Doubling	1-letter consonant doubled before *–ing*/*-ed*	beg/begging run/running
E deletion	Silent *e* dropped before *–ing* and *-ed*	make/making
E insertion	*e* added after *–s*, *-z*, *-x*, *-ch*, *-sh* before *-s*	miss/misses watch/watches mash/mashes
Y replacement	*-y* changes to *–ie* before *–s*, *-i* before *-ed*	try/tries
K insertion	Verbs ending in vowel + *-c* add *-k*	panic/panicked

Multi-Tape Machines

- To deal with this we can simply add more tapes and use the output of one tape machine as the input to the next
- So to handle irregular spelling changes we'll add intermediate tapes with intermediate symbols

Multi-Level Tape Machines



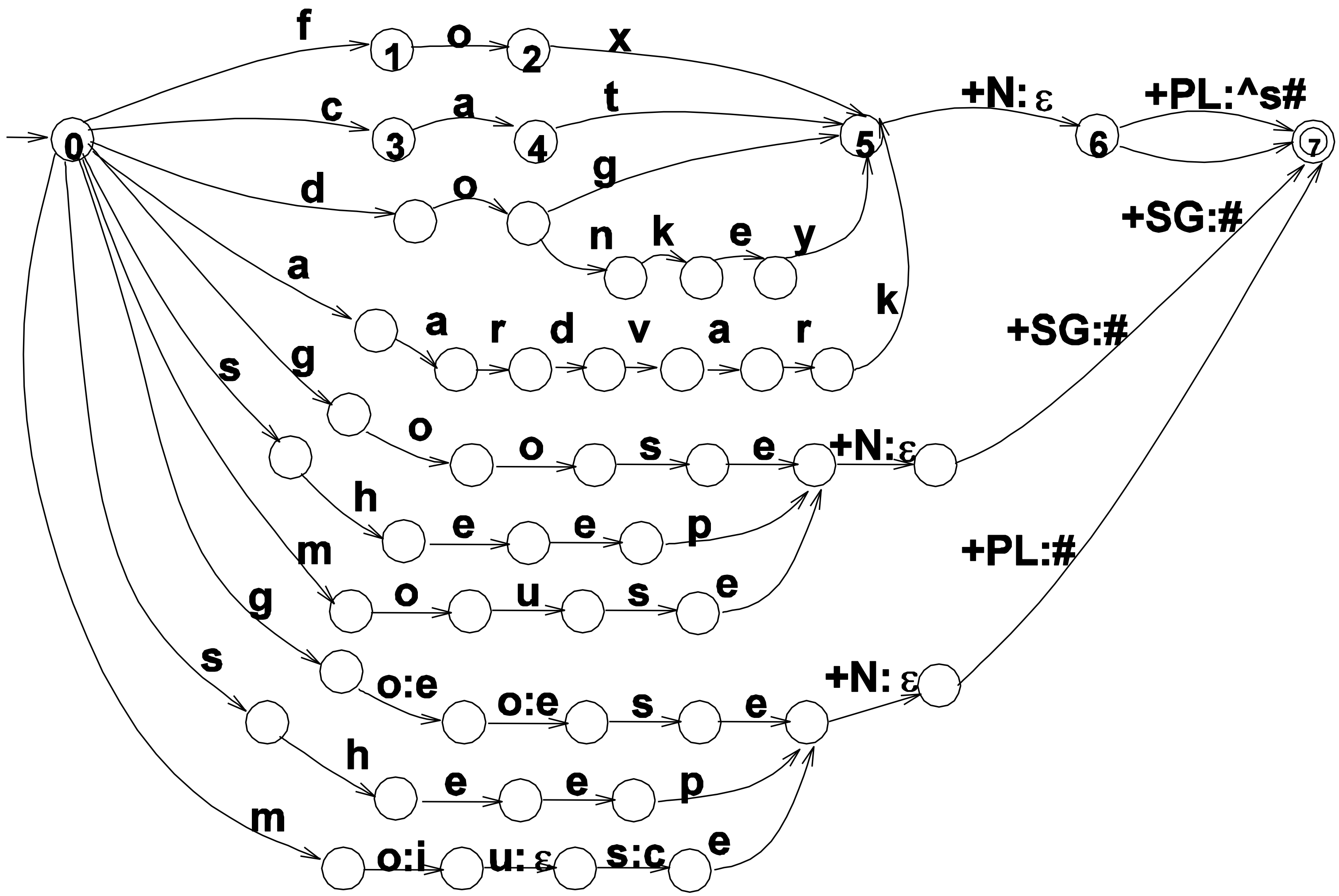
- We use one machine to transduce between the lexical and the intermediate level, and another to handle the spelling changes to the surface tape
- ^ is morpheme boundary; # is word boundary

Orthographic Rules and FSTs

- Define additional FSTs to implement rules such as consonant doubling (beg → begging), 'e' deletion (make → making), 'e' insertion (watch → watches), etc.

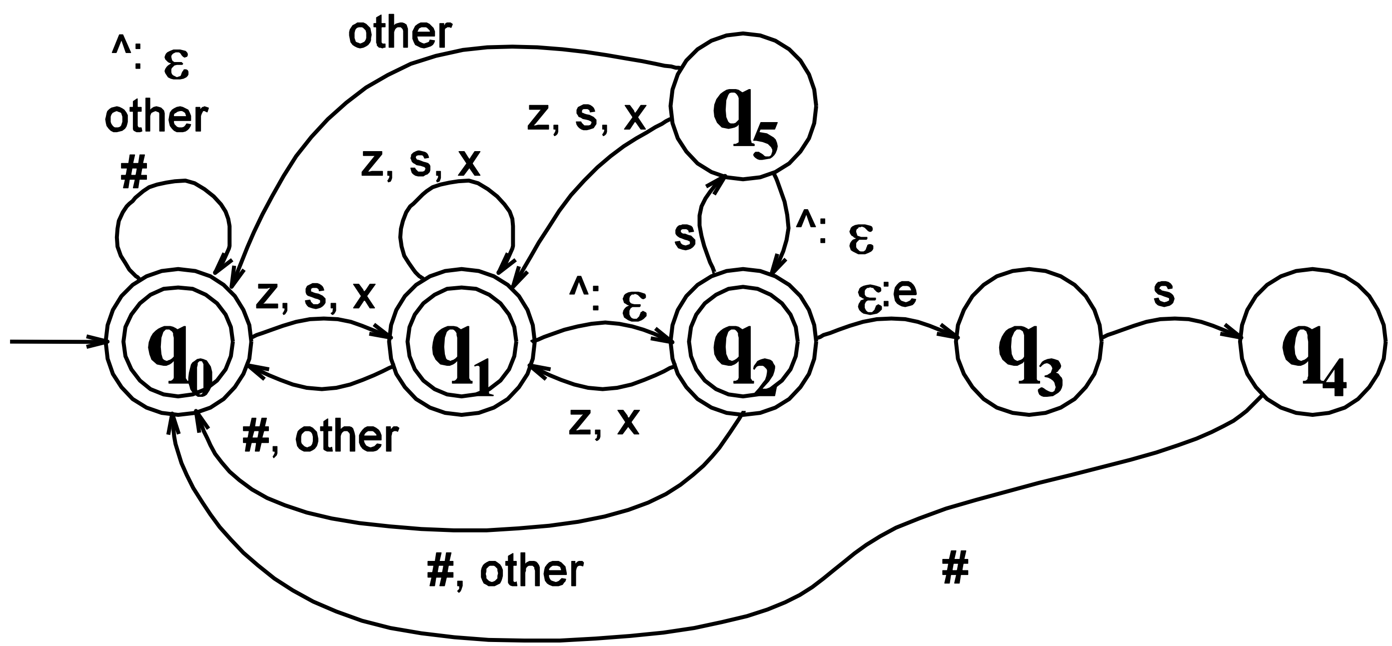
Lexical	f	o	x	+N	+PL	
Intermediate	f	o	x	^	s	#
Surface	f	o	x	e	s	

Lexical to Intermediate Level



Intermediate to Surface

- The add an “e” rule as in `fox^s# <-> foxes`



Note

- A key feature of this machine is that it doesn't do anything to inputs to which it doesn't apply.
- Meaning that they are written out unchanged to the output tape.

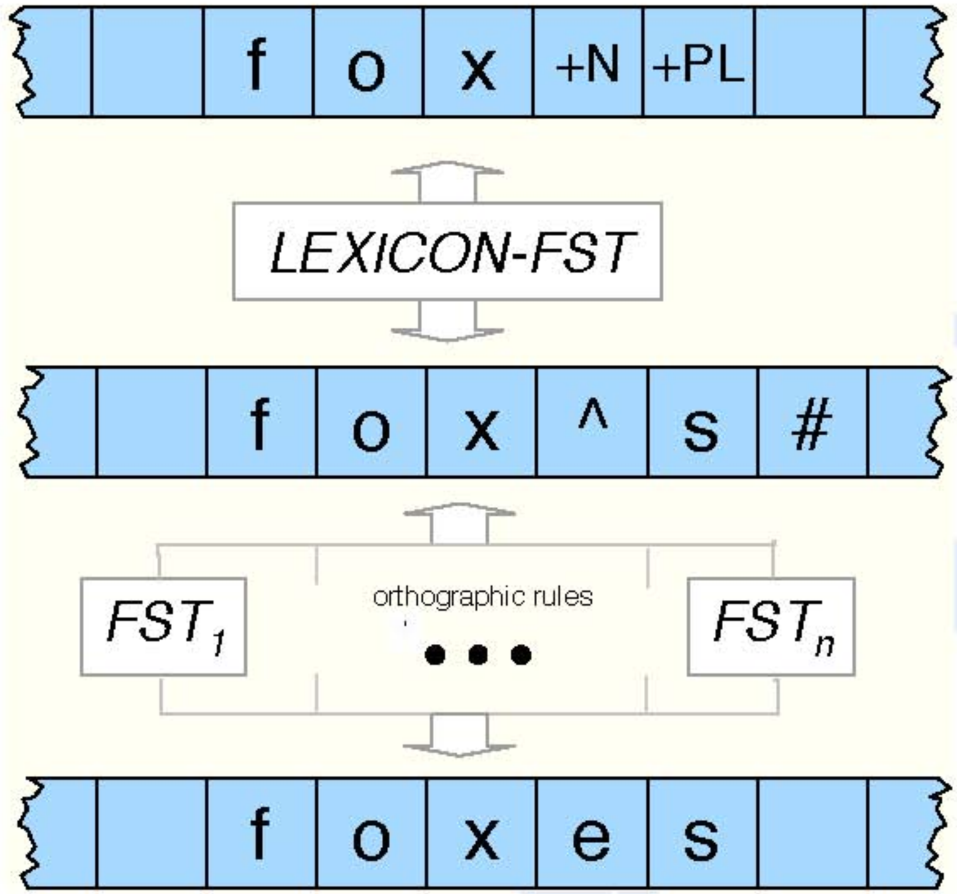


Figure 3.19 Generating or parsing with FST lexicon and rules

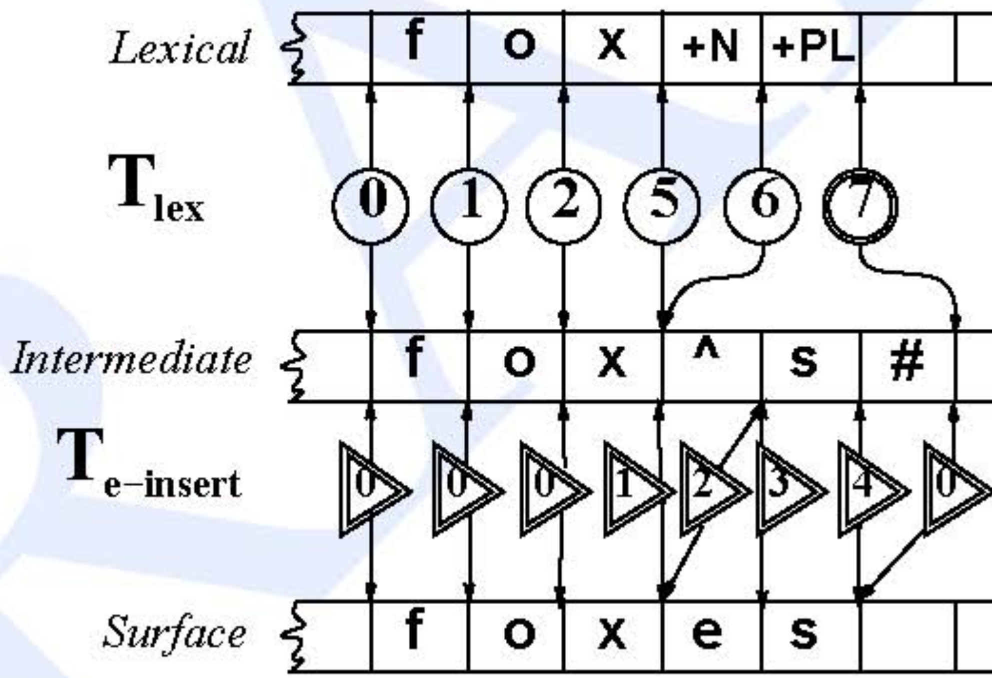


Figure 3.20 Accepting *foxes*: The lexicon transducer T_{lex} from Fig. 3.14 cascaded with the E-insertion transducer in Fig. 3.17.

- Note: These FSTs can be used for generation as well as recognition by simply exchanging the input and output alphabets (e.g. ^s#:+PL)

Summing Up

- FSTs provide a useful tool for implementing a standard model of morphological analysis, Kimmo's two-level morphology
  - Key is to provide an FST for each of multiple levels of representation and then to combine those FSTs using a variety of operators (cf AT&T FSM Toolkit)
  - Other (older) approaches are still widely used, e.g. the rule-based Porter Stemmer