Introduction to Natural Language Processing

Chapter 3

Outline

Administration

- Homework 1 Clarification
- Reading List
- Syllabus

Demo Debriefing

Review FSAs

English Morphology

FSA and Lexicons

Finite State Transducers

Porter Stemmer

Soundex

Human Morphology

Review of FSAs

FSAs are formal devices used to recognize strings in a language.

FSAs can be specified using any of a number of formalisms.

FSAs can be combined in various ways to create FSAs for new regular languages.

Morphology

Morphology is the study of the way words are built up from smaller meaningful units called morphemes.

We can usefully divide the world of morphemes into two broad classes:

- stems - the core meaningful units in a lexicon
- affixes - bits and pieces that combine with stems to modify their meanings and grammatical functions
- examples
  - *im*material, try*ing*, *un*read*able*
- note: morpheme does not equal syllable (*singer* vs. *finger*, *sing* vs. *sang*)

Full Listing vs. Minimal Redundancy

- *true, truer, truest, truly, untrue, truth, thruthful, truthfully, untruthfully, untruthfulness*
- *untruthfulness* = *un-* + *true* + *-th* + *-ful* + *-ness*
- by representing knowledge about the internal structure of words and the rules of word formation, we don't need to list each related word separately (10 entries) in our lexicon

Affixes

Prefixes

- precede the stem
- *unbuckle*

Suffixes

- follow the stem
- *eats*

Infixes

- inserted inside the stem
- *absobl**dylutely*

Circumfixes

- precede and follow the stem
- not in English

Morphological Parsing

Parsing

- producing some structure for an input
- will be relevant for many levels of knowledge

Morphological Parsing (or Stemming)

- taking a surface <u>input</u> and breaking it down into its morphomes
- *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)

Why not instead include all words in a dictionary (e.g., all plurals)?

- inefficient because rules apply to many words
- rules apply to new words
- other languages besides English

Applications include IR, MT, spell checkers

Morphology (continued)

And we can also usefully divide morphology into two broad classes:

- inflectional morphology
- derivational morphology

Word Class

By word class we have in mind notions like noun and verb with which you are probably familiar.

We'll go into the gory details later (Chapter 8). For now it is sufficient to know about nouns, verbs, and adjectives.

We're concerned with word classes because the way that stems combine with affixes to form words is to a large degree based on the stem's word class.

Inflectional Morphology

Inflectional morphology concerns the combination of a stem with a grammatical morpheme which more or less results in a word...

- with the same word class as the stem
- which can serve a grammatical role that the stem could not (e.g., agreement)
- bird $\rightarrow$ bird$s$
- like $\rightarrow$ like$s$ or liked

Nouns and Verbs (in English)

English is simple.

- only nouns, verbs, and sometimes adjectives can be inflected
- small number of possible inflectional affixes

Nouns have a simple inflectional morphology: markers for *plural* and markers for *possessives*.

Verbs are slightly more complex: markers appropriate to the function the verb is being asked to serve.

Regulars and Irregulars

Morphology is complicated by the fact that many words refus to follow the rules.

- mouse/mice, ox/oxen, goose/geese
- go/went, fly/flew

We'll use the terms regular and irregular to refer to words that follow the rules and words that don't.

Nouns

Regular Nouns

- singular (cat, thrush)
- plural (cats, thrushes)
- possessive (cat's, thrushes')

Irregular Nouns

- singular (mouse, ox)
- plural (mice, oxen)

Verbal Inflection

English is much simpler than many languages

See tables on pages 62 and 63 in Jurafsky and Martin

Most verbs are relatively regular

- stem: infinitive (I want to walk home), after certain other verbs (I'd rather walk home)
- -*s* form: habitual present (She walks every Wednesday)
- -*ing* participle: gerund (Walking is fine if you have a sidewalk)
- -*ed*: past form (I walked home yesterday)

But what about *eat, ate, eaten* or *catch, caught*?

Regular Verbs

If you know a regular verb stem, you can predict the other forms by adding one of the predictable endings and making some regular spelling changes.

Regular verbs cover a large majority of verbs.

The regular class is *productive* (includes new verbs).

- *emailed, instant-messaged, faxed, homered*

Irregular Verbs

Three to eight morphological forms in English.

Small number of verbs (around 250), but among the most frequent.

Primary (*be, have, do*) and modal verbs (*can, will, must*) are often irregular and not productive

- *am, is, are, were, was, been, being*

Derivational Morphology

Derivational morphology refers to the stuff that you know but no one ever told you.

Characterized by

- quasi-systematicity
- irregular meaning changes
- changes of word class

See tables on page 64 in Jurafsky and Martin

Derivational Morphology Examples

Nominalization (Verbs becomes Nouns)

- -*ize* verbs $\rightarrow$ -*ation* nouns
- $generalize, realize \rightarrow generalization, realization$

Verbs, Nouns become Adjectives

- $embrace, pity \rightarrow embraceable, pitiable$
- $care, wit \rightarrow careless, witless$

Adjectives become Adverbs

- $happy \rightarrow happily$

English is complex

More complicated to model than inflection

- less productive: **scienceless, concernless, goable, sleep-able*
- meanings of derived terms harder to predict by rule: *clueless, careless*

Finite State Morphological Parsing

Inputs

- cats
- goose
- geese
- gooses

Morphologically parsed output (stem and <u>features</u>)

- cat +N +PL
- (goose +N +SG) or (GOOSE +v) (ambiguous)
- (goose +N +PL)
- (goose +V +3SG)

Alternative bracketing representation

- indecipherable → [in [[de [cipher]] able]]
- uncontrollable → [[un- [[control]V + -able]ADJ]ADJ -ly]ADV

Outline

Homework questions?

Review (English) Morphology

FSA and Lexicons

Finite State Transducers

Porter Stemmer

Soundex

Human Morphology

Review of Morphology

Morphology - the way words are built up from smaller meaningful units called morphemes

- stems - the core meaningful units in a lexicon
- affixes - bits and pieces that combine with stems to modify their meanings and grammatical functions (prefixes, suffixes, infixes)

Inflectional Morphology

Derivational Morphology

Morphological Parsing Requirements

Lexicon

- word repository
- stems and affixes (with corresponding parts of speech)

Morphotactics

- morpheme ordering
- model of how morphemes can be affixed to a stem

Orthographic Rules

- spelling changes due to affixation

Using FSAs to Represent a Lexicon

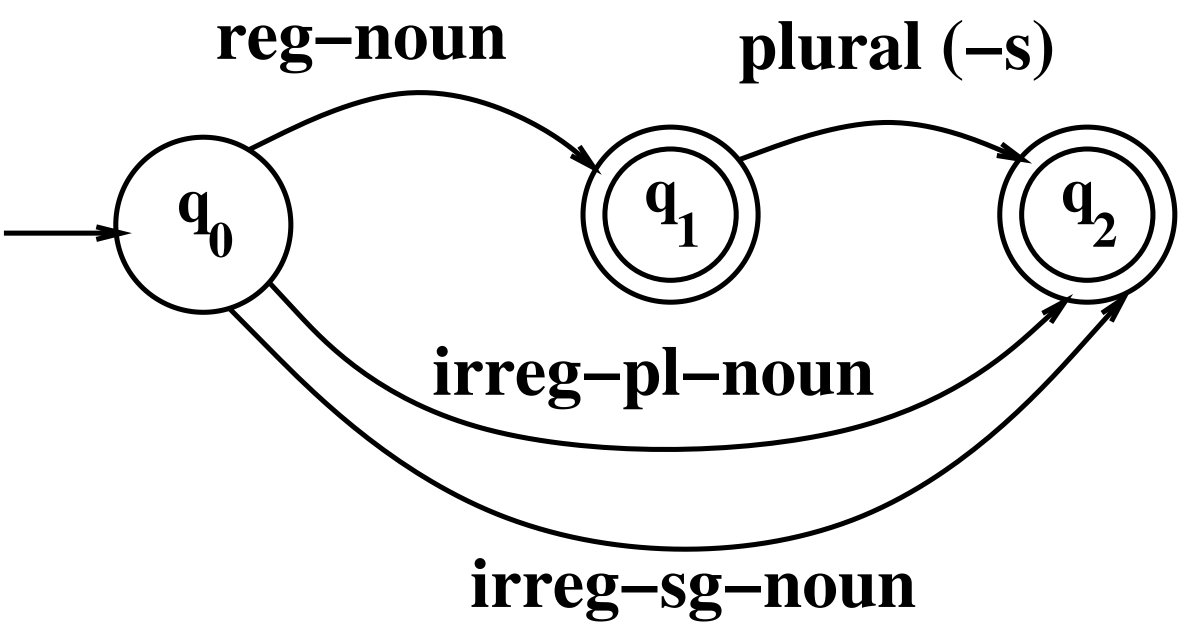
First we'll try to capture the morphotactics.

- the model of morpheme ordering
- e.g., the plural morpheme follows nouns in English

Then we'll layer in the actual words in the lexicon.

- the list of stems and affixes, together with basic information such as stem class
- an alternative to exhaustive lists, which are often inconvenient or impossible

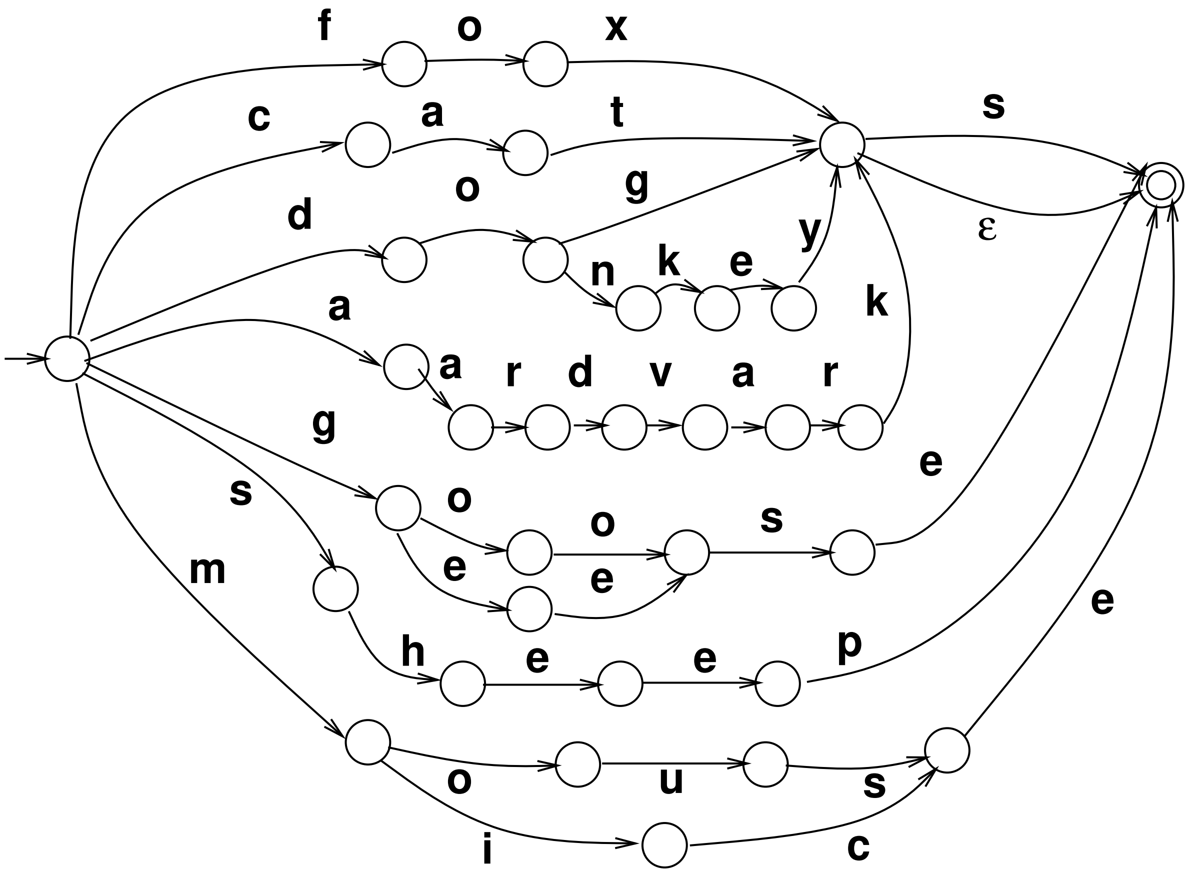
Capturing Morphotactics using FSA



English nominal inflection

Inputs: cats, goose, geese

Embedding the Lexicon



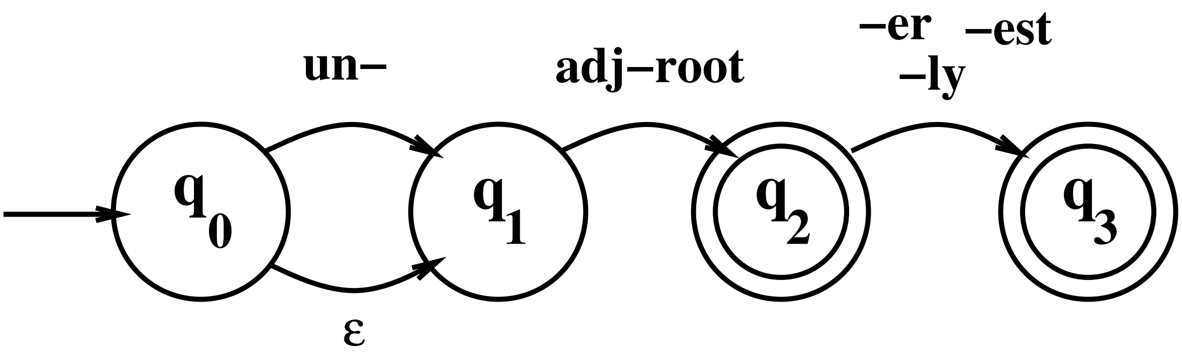
We can expand each non-terminal in our FSA into each stem in its class

- e.g., reg-n = cat, dog . . .

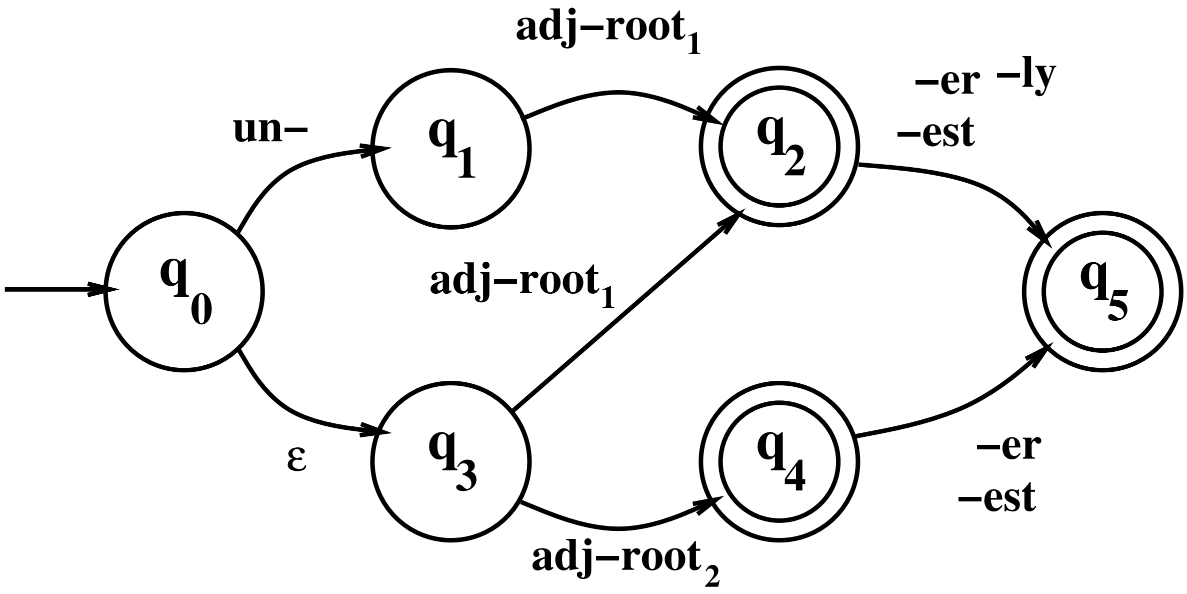
and then expand each such stem to the letters it includes

- e.g., cat $\rightarrow$ c a t

English Adjective Fragment FSA



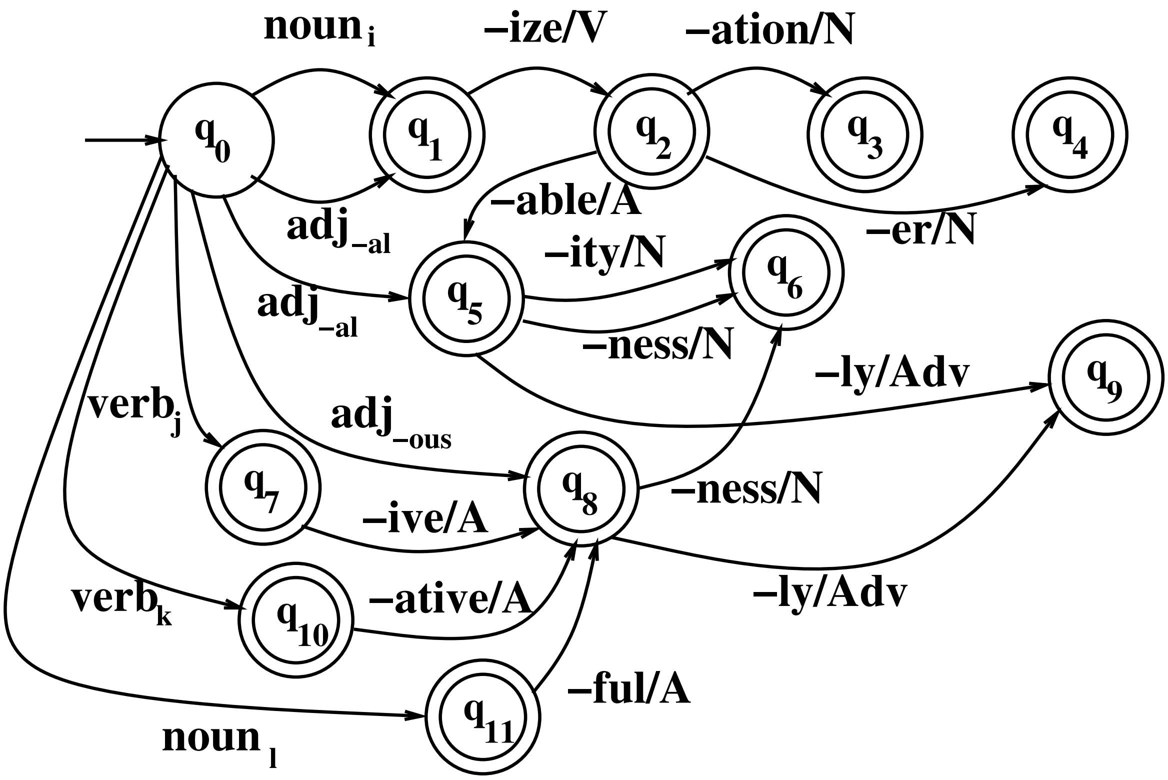
Second Draft



Inputs: clear, happy, real, big, red

How do we embed the lexicon, e.g., big, red?

Modeling Derivational Facts



Examples

- fossilize → fossilization
- equal → equality

Who cares?

Should search engines make use of morphological information?

- how might they make use of it?
- what happens if they do?
- what happens if they don't?

Who cares?

Spell checking

- is *muncheable* a legal word?

To identify part of speech (chapter 8)

- is *goose* a noun or a verb?
- useful for parsing, machine translation

To identify a word's stem

- search engines, information retrieval (don't need to type in both *cat* and *cats*)

Limitations

To cover a language (e.g., English) will require very large FSAs with consequent search problems

- adding new items to the lexicon means recomputing the FSA

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 of what sort?
- we used this information to build our FSA, can we get it back?

Recognition versus Parsing

Morphological Recognition determines whether an input string of letters makes up a legitimate word by embedding the lexicon in a morphotactic FSA.

Morphological Parsing is the process of taking an input word and creating a structure for it.

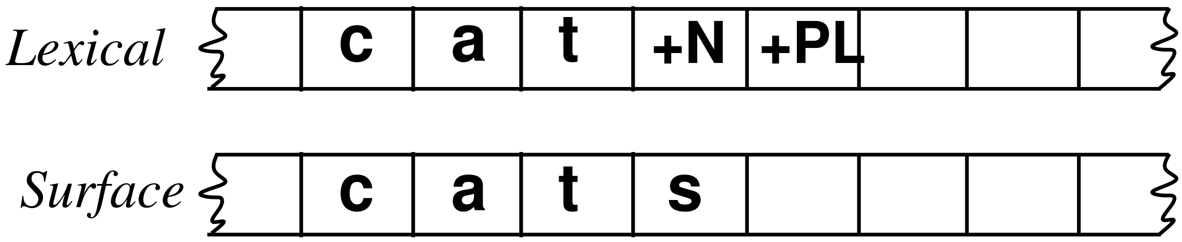
- cats $\rightarrow$ cat +N +PL

Finite State Transducers (FST)

The simple story . . .

- add an extra tape to normal FSA
- add extra symbols to the transitions of a normal FSA
- i.e., one tape has `cats` on it and we write `cat+N+PL` on the other

Tapes and transitions



Two Level Morphology represents words as a correspondence between lexical (the morphemes) and surface (the orthographic word) levels.

Parsing rules maps surface to lexical level.

We usually visualize a FST as a 2-tape FSA which recognizes/generates *pairs* of strings.

FSTs (continued)

FSTs map between one set of symbols and another using an FSA whose alphabet is composed of pairs of symbols from input and output alphabets.

In the typical use of FSTs we'll

- read from one tape (just as in FSA recognition), using the second symbol on each transition,
- and write to the second tape, using the first symbol on each transition.

In general, FSTs can be used for

- translators (Hello:Ciao)
- parser/generators (Hello:How may I help you?)
- 2-level morphology

FSTs and Ambiguity

Recall that in ND-FSA recognition, there can be multiple paths through a machine leading to an accept condition. Doesn't matter which is found.

In FSTs, the path taken through the machine actually matters.

Parsing *unionizable*

- `union-ize-able`
- `un-ion-ize-able`
- each represents a valid path through the machine resulting in a different parse.

FSTs and Ambiguity (continued)

There are a number of ways to handle this problem.

- simply accept the first successful structure written on the tape
- run the parse through all possible paths writing out all possible valid structures along the way
- bias that search in some manner so that only one (or a few) valid paths are actually pursued

Much of NLP involves implementing the last option.

FSTs and Ambiguity (continued)

Unionizable represents a problem involving a global ambiguity.

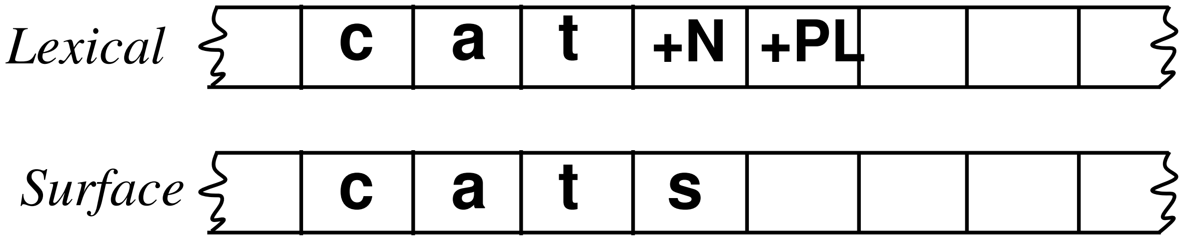
What's the problem with parsing *assess*?

Review

FSAs for morphological recognition

- lexicon representation

FSTs for morphological parsing



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

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

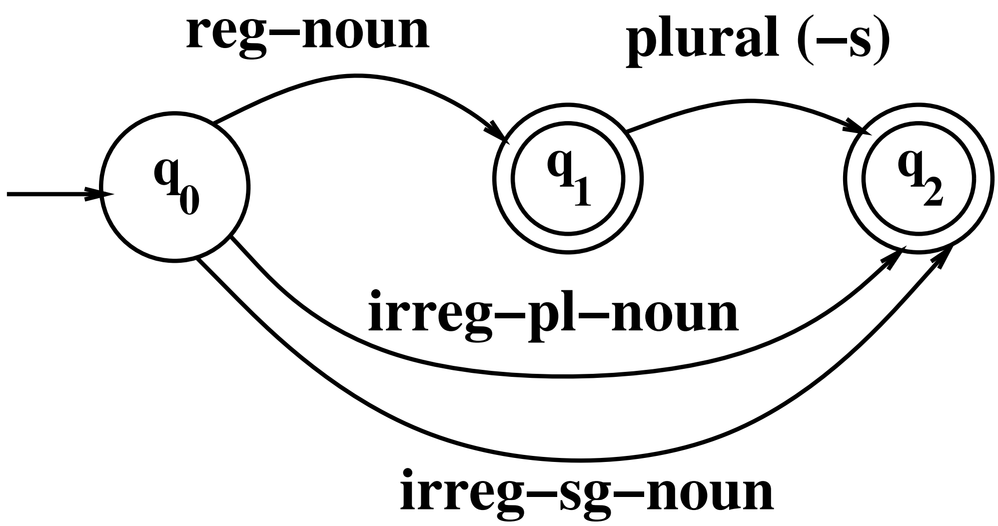
Example

Draw a FST to translate from our sheep language to a quizzical cow language, e.g.,

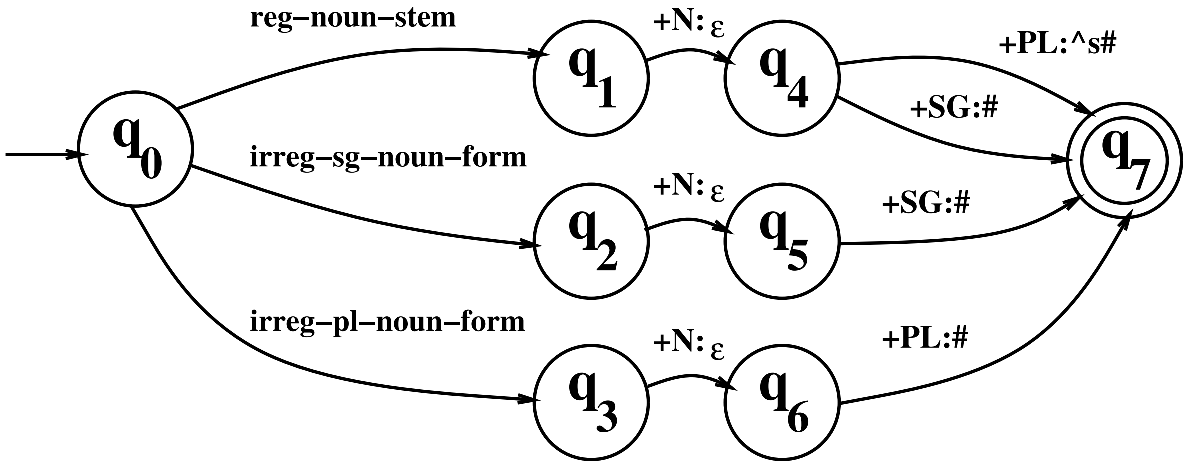
- $baaa! \rightarrow mooo?$

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.

Cascading (combining, composing) replaces e.g. reg-n with every regular noun representation in the lexicion.

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

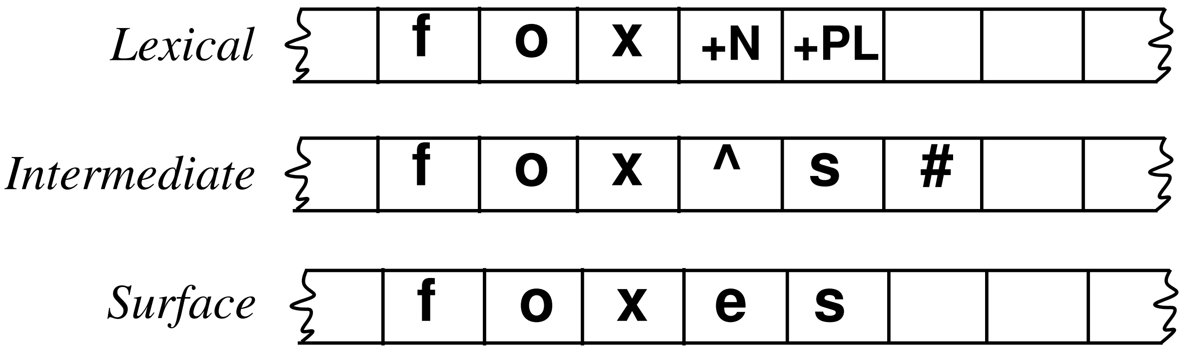
See table on page 77 of Jurafsky and Martin

Define additional FSTS to implement spelling rules, e.g.,

- consonant doubling ($beg \rightarrow begging$)
- ‘e’ deletion ($make \rightarrow making$)
- ‘e’ insertion ($watch \rightarrow watches$)

Multi-Level Multi-Tapes (cont.)

So to handle spelling we use three tapes:

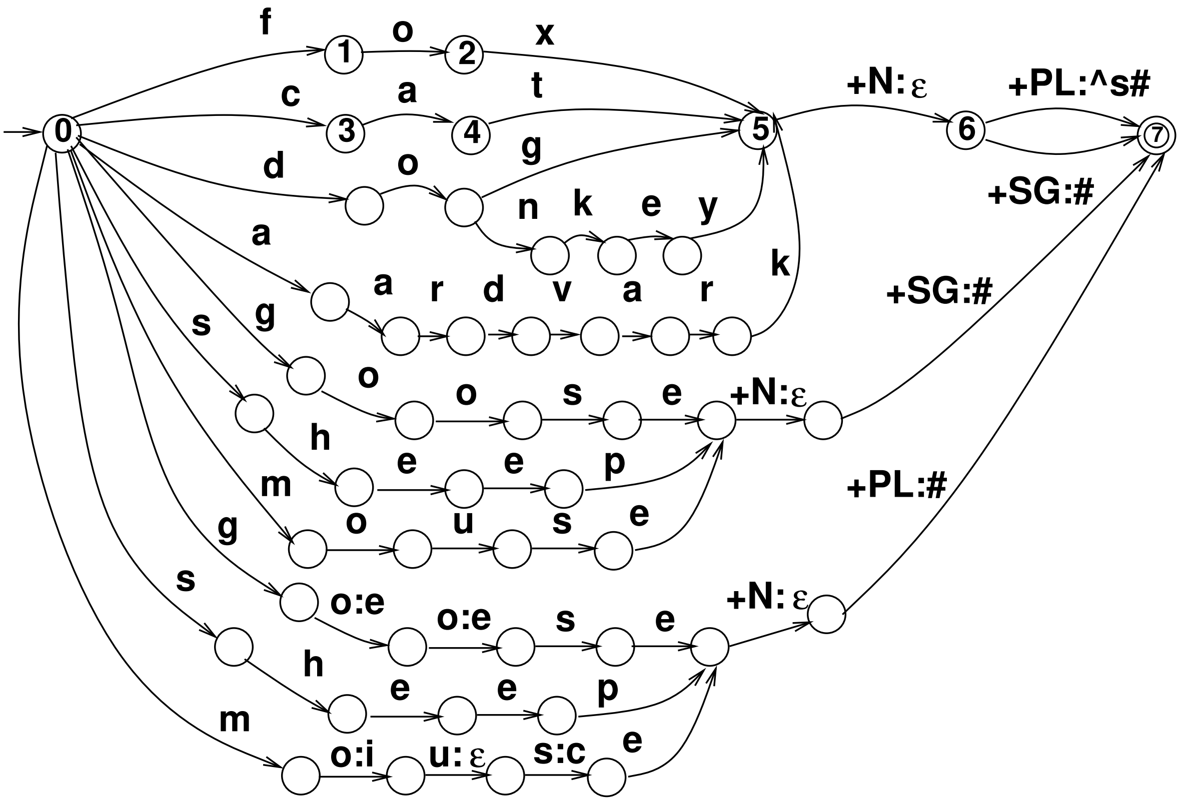


We need one transducer to work between the lexical and intermediate levels and a second (bunch) to work between the intermediate and surface levels to patch up the spelling (e.g., to do e-insertion).

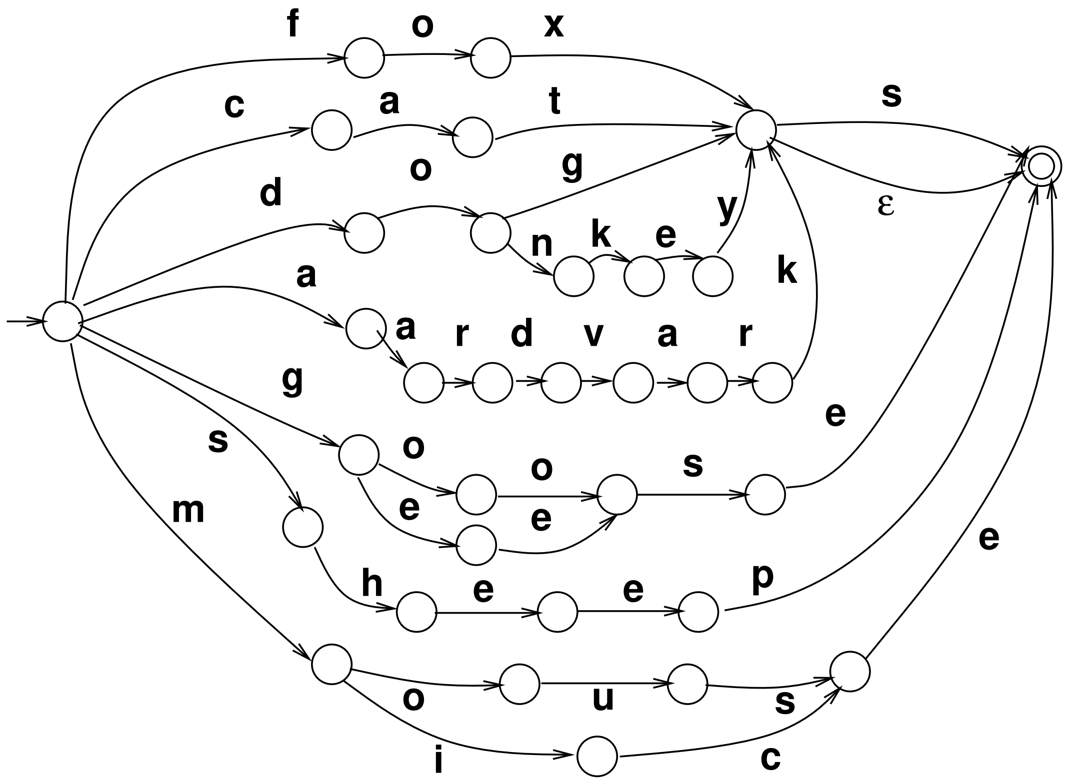
Intermediate tapes: tapes with boundary markers

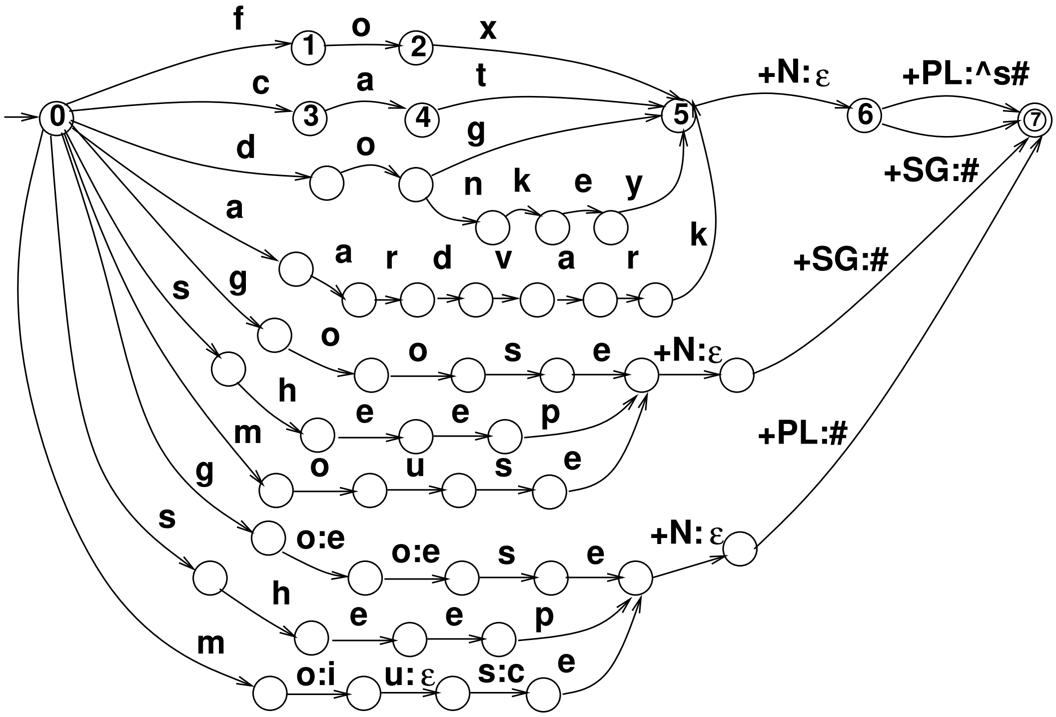
Lexical to Intermediate Transducer

After composition...



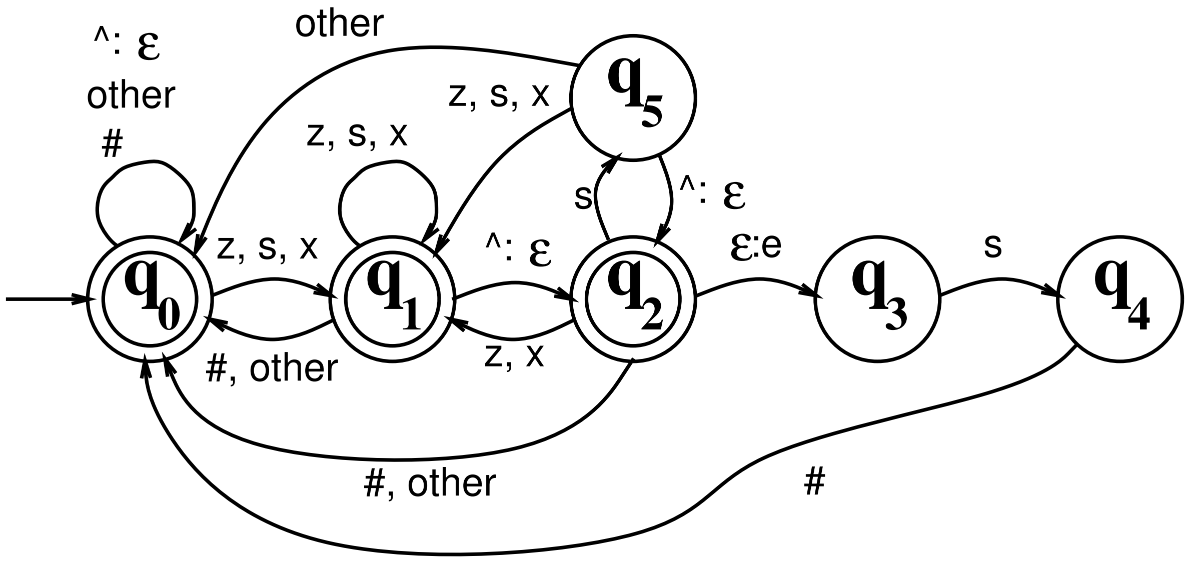
Nominal Inflection (FSA vs. FST)





Intermediate to Surface Transducer

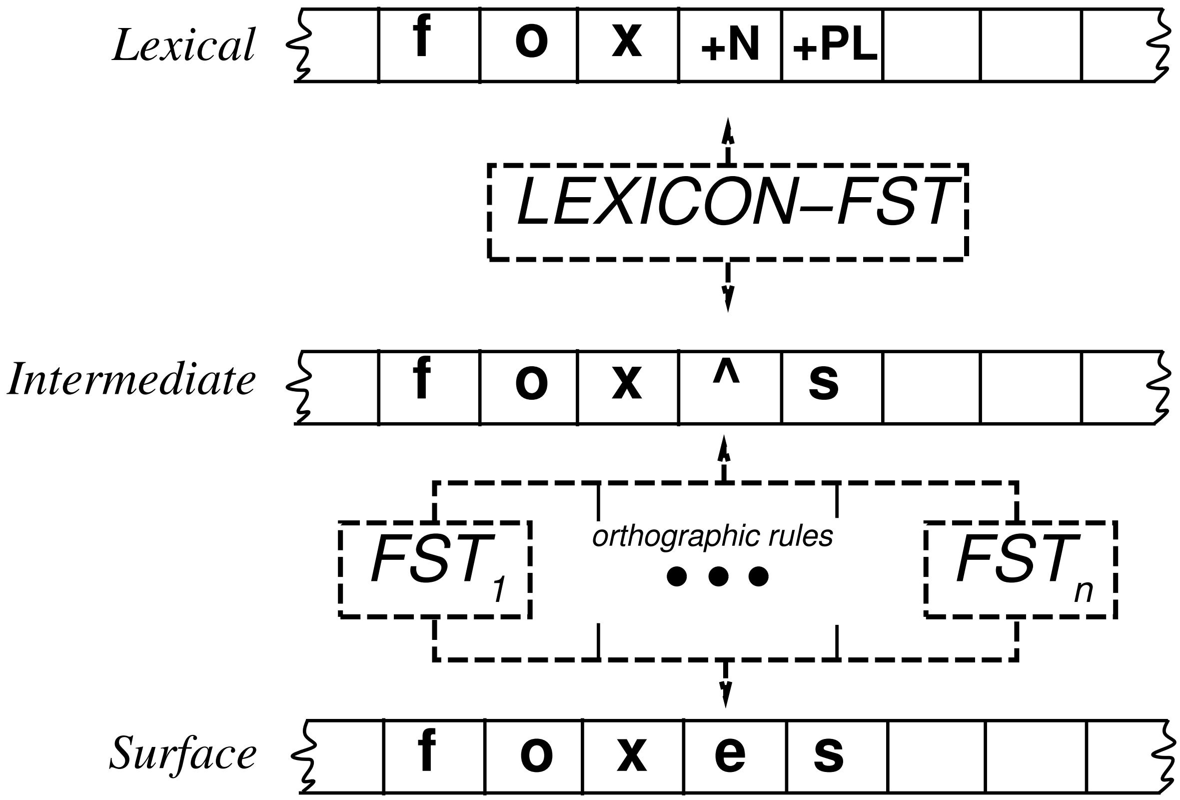
We need an FST to add an e between an x, s or z and before the s at morpheme boundary at the end of a word.



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

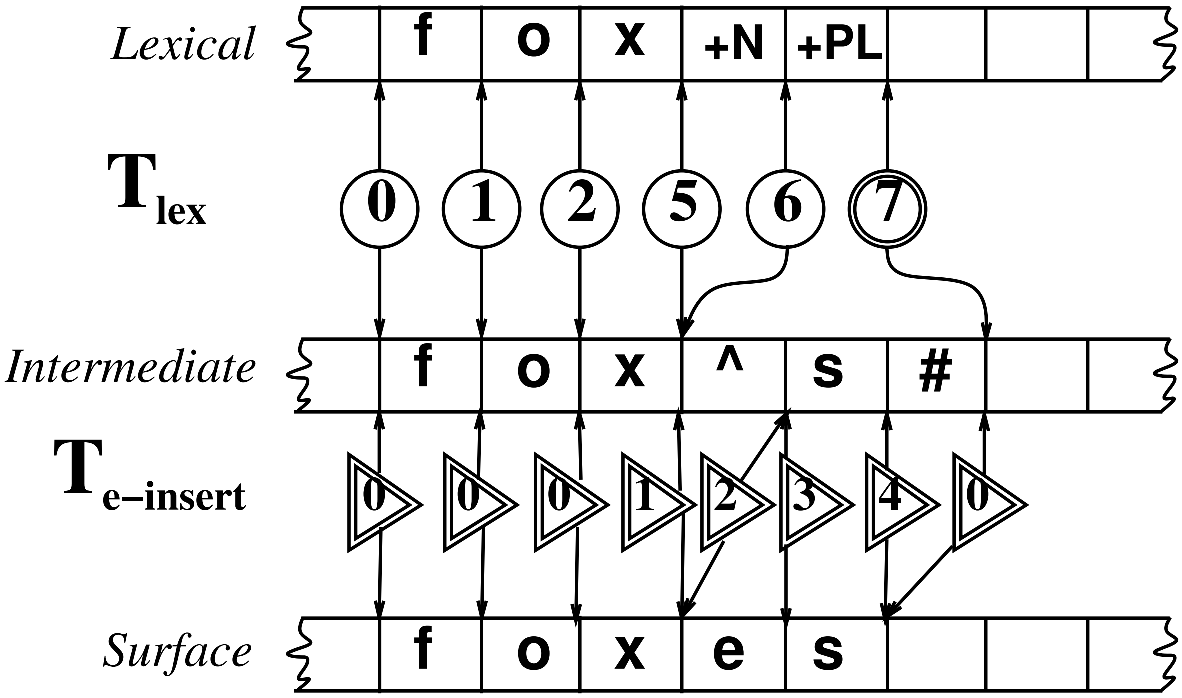
- generation: use top-down
- parsing: use bottom-up
- FSTs can be used for generation and recognition by simply exchanging the input and output alphabets

  `^s#:+PL`

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)

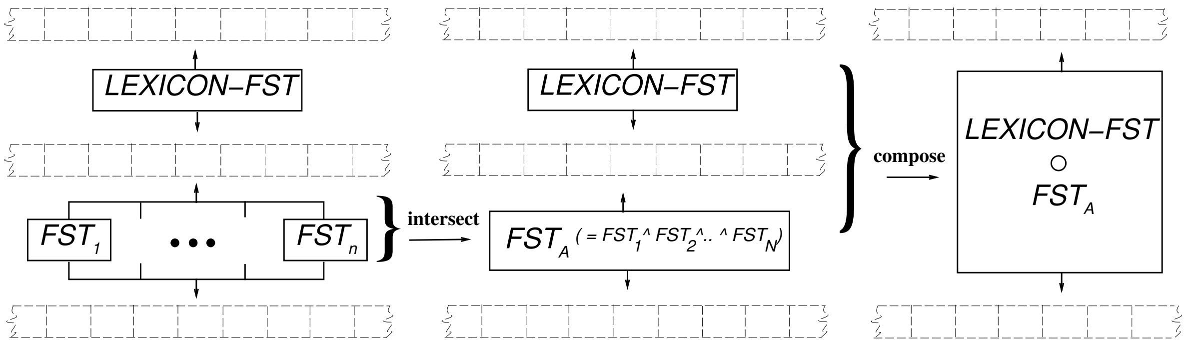
Composition Construction

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...



<u>Compose</u> a cascade of transducers in series into a single more complex transducer

Transducers in parallel can be combined by <u>intersection</u>.

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 <u>Stemming</u>.

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")

Summing Up

FSTs provide a useful tool for implementing a standard model of morphological analysis (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

Toolkits such as AT&T FSM Toolkit available

Other approaches are also still used, e.g., the rule-based Porter Stemmer

For Next Time

Chapter 6

For Fun: Find a non-native English speaker (perhaps yourself) to examine inflection in another language