Morphological analysis

LING 570

Fei Xia

Outline

- The task
- Porter stemmer
- FST morphological analyzer: J&M-ed2 3.1-3.8

The task

- To break a word down into component morphemes and build a structured representation

- A morpheme is the minimal meaning-bearing unit in a language.
  - Stem: the morpheme that forms the central meaning unit in a word
  - Affix: prefix, suffix, infix, circumfix
    - Prefix: e.g., possible ➔ impossible
    - Suffix: e.g., walk ➔ walking
    - Infix: e.g., hingi ➔ humingi (Tagalog)
    - Circumfix: e.g., sagen ➔ gesagt (German)

Two slightly different tasks

- Stemming:
  - Ex: writing ➔ writ + ing (or write + ing)

- Lemmatization:
  - Ex1: writing ➔ write +V +Prog
  - Ex2: books ➔ book +N +Pl
  - Ex3: writes ➔ write +V +3Per +Sg

Ambiguity in morphology

- flies ➔ fly +N +PL
- flies ➔ fly +V +3rd +Sg

- saw ➔ see +V +past
- saw ➔ saw +N

Language variation

- Isolated languages: e.g., Chinese

- Morphologically poor languages: e.g., English

- Morphologically complex languages: e.g., Turkish

Ways to combine morphemes to form words

- Inflection: stem + gram. morpheme ➔ same class
  - Ex: help + ed ➔ helped
- Derivation: stem + gram. morpheme ➔ different class
  - Ex: civil + -zation ➔ civilization
- Compounding: multiple stems
  - Ex: cabdriver, doghouse, waterfront
- Cliticization: stem + clitic
  - Ex: they’ll, she’s (“she is” vs. “she has”)

Porter stemmer

Porter stemmer

- The algorithm was introduced in 1980 by Martin Porter.
- http://www.tartarus.org/~martin/PorterStemmer/def.txt
- Purpose: to improve IR.
- It removes suffixes only.
  - Ex: civilization ➔ civil
- It is rule-based, and does not require a lexicon.

How does it work?

- The format of rules: (condition) S1 ➔ S2

  Ex: (m>1) ZATION ➔ ϵ

- Rules are partially ordered:
  - Step 1a: -s
  - Step 1b: -ed, -ing
  - Step 2-4: derivational suffixes
  - Step 5: some final fixes

- How well does it work? What are the main problems with this kind of approach?

FST morphological analyzer

English morphology

- Affixes: have prefixes and suffixes, but no infixes, circumfixes.
- Inflectional:
  - Noun: -s
  - Verbs: -s, -ing, -ed, -ed
  - Adjectives: -er, -est
- Derivational:
  - Ex: V + suffix ➔ N

    computerize + -ation ➔ computerization

    kill + er ➔ killer
- Compound: pickup, database, heartbroken, etc.
- Cliticization: ‘m, ‘ve, ‘re, etc.

➔ For now, we will focus on inflection only.

FST morphological analysis

- Read J&M-ed2 Chapter 3

- English morphology:

- FSA acceptor:
  - Ex: cats ➔ yes/no, foxs ➔ yes/no

- FSTs for morphological analysis:
  - Ex: fox +N +PL ➔ fox^s#

- Adding orthographic rules: (see additional slides)
  - Ex: fox^s# ➔ foxes#

Three components

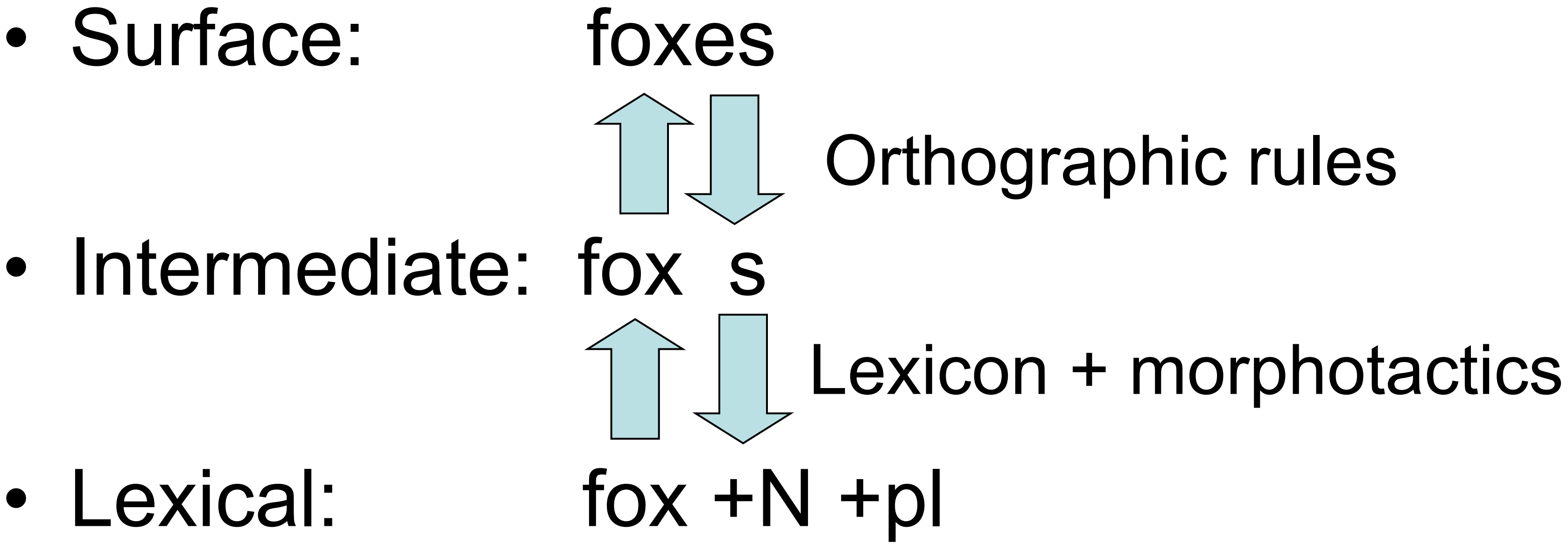
- Lexicon: the list of stems and affixes, with associated features.
  - Ex: book: N
    -s: +PL

- Morphotactics:
  - Ex: +PL follows a noun

- Orthographic rules (spelling rules): to handle exceptions that can be dealt with by rules.
  - Ex1: y ➔ ie: fly + -s ➔ flies
  - Ex2: ϵ ➔ e: fox + -s ➔ foxes
  - Ex2’: ϵ ➔ e / x^_s#

An example

- Task: foxes → fox +N +PL



- Surface: foxes

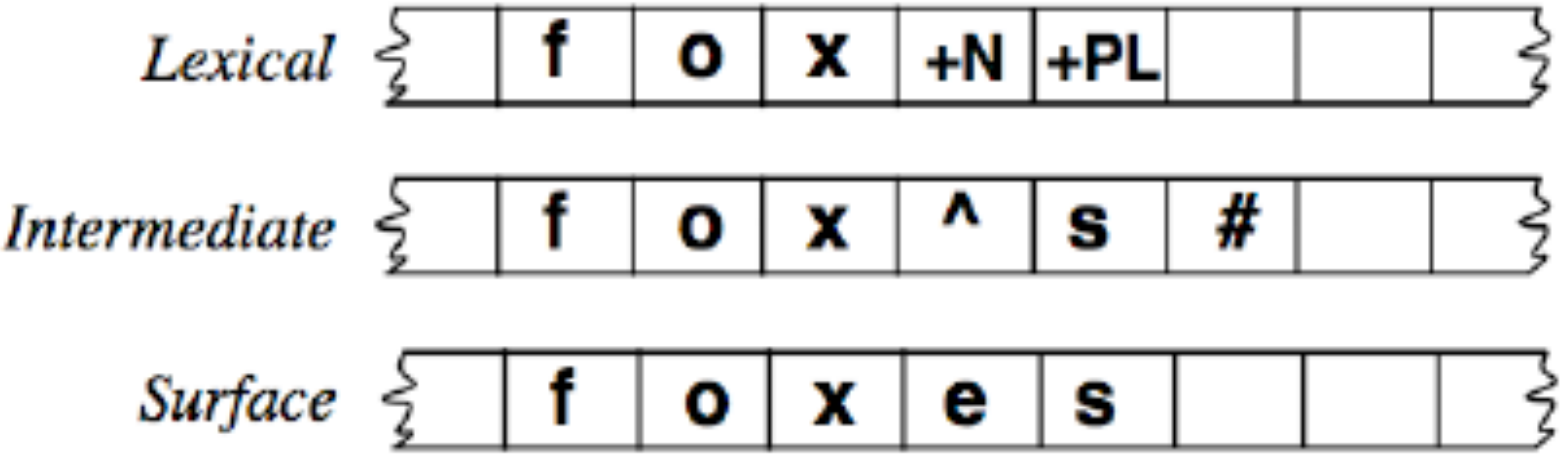
Orthographic rules

- Intermediate: fox s

Lexicon + morphotactics

- Lexical: fox +N +pl

Three levels



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

analysis: foxes => fox^s#   fox^s# => fox +N +PL

generation: fox +N +PL => fox^s#   fox^s# => foxes

The lexicon (in general)

- The role of the **lexicon** is to associate linguistic information with words of the language.

- Many words are ambiguous: with more than one entry in the lexicon.

- Information associated with a word in a lexicon is called a **lexical entry**.

What is in a lexicon?

- fly: v, +base
- fly: n, +sg
- fox: n, +sg

- fly: (NP, V)
- fly: (NP, V, NP)

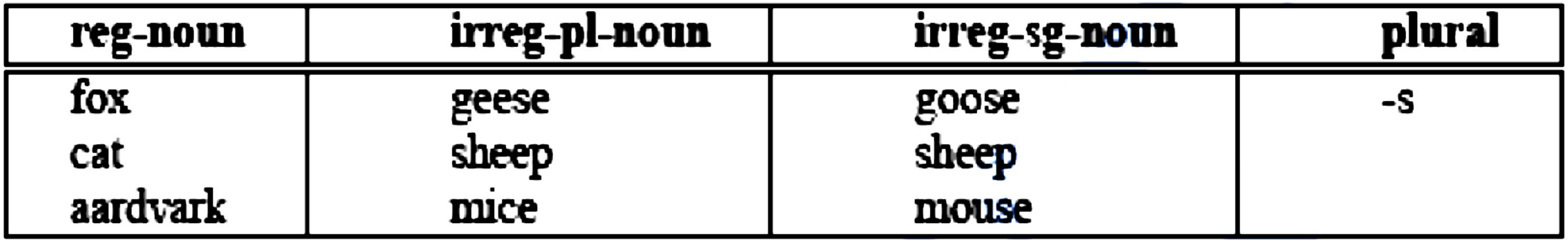
Should the following be included in the lexicon?

- flies: v, +sg +3rd
- flies: n, +pl
- foxes: n, +pl

- flew: v, +past

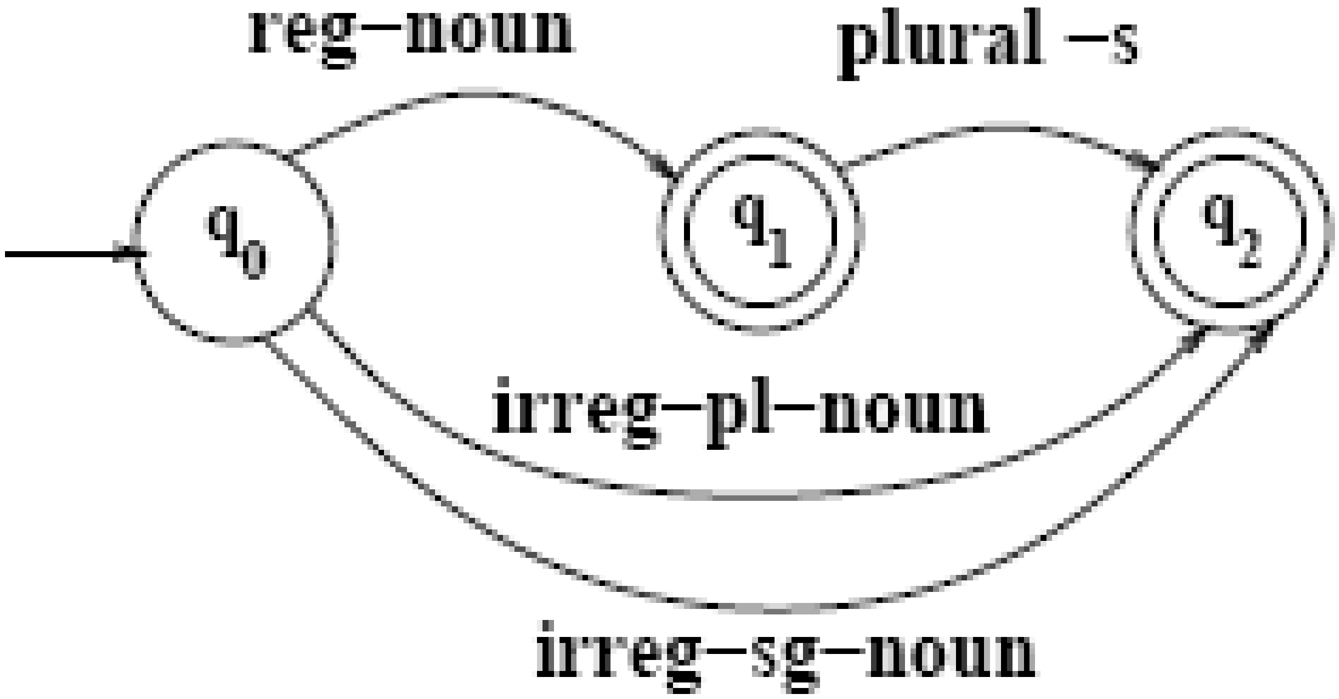
The lexicon for English noun inflection

- fox: n, +sg, +reg ⬄ reg-noun
- goose: n, +sg, -reg ⬄ irreg-sg-noun
- geese: n, +pl, -reg ⬄ irreg-pl-noun

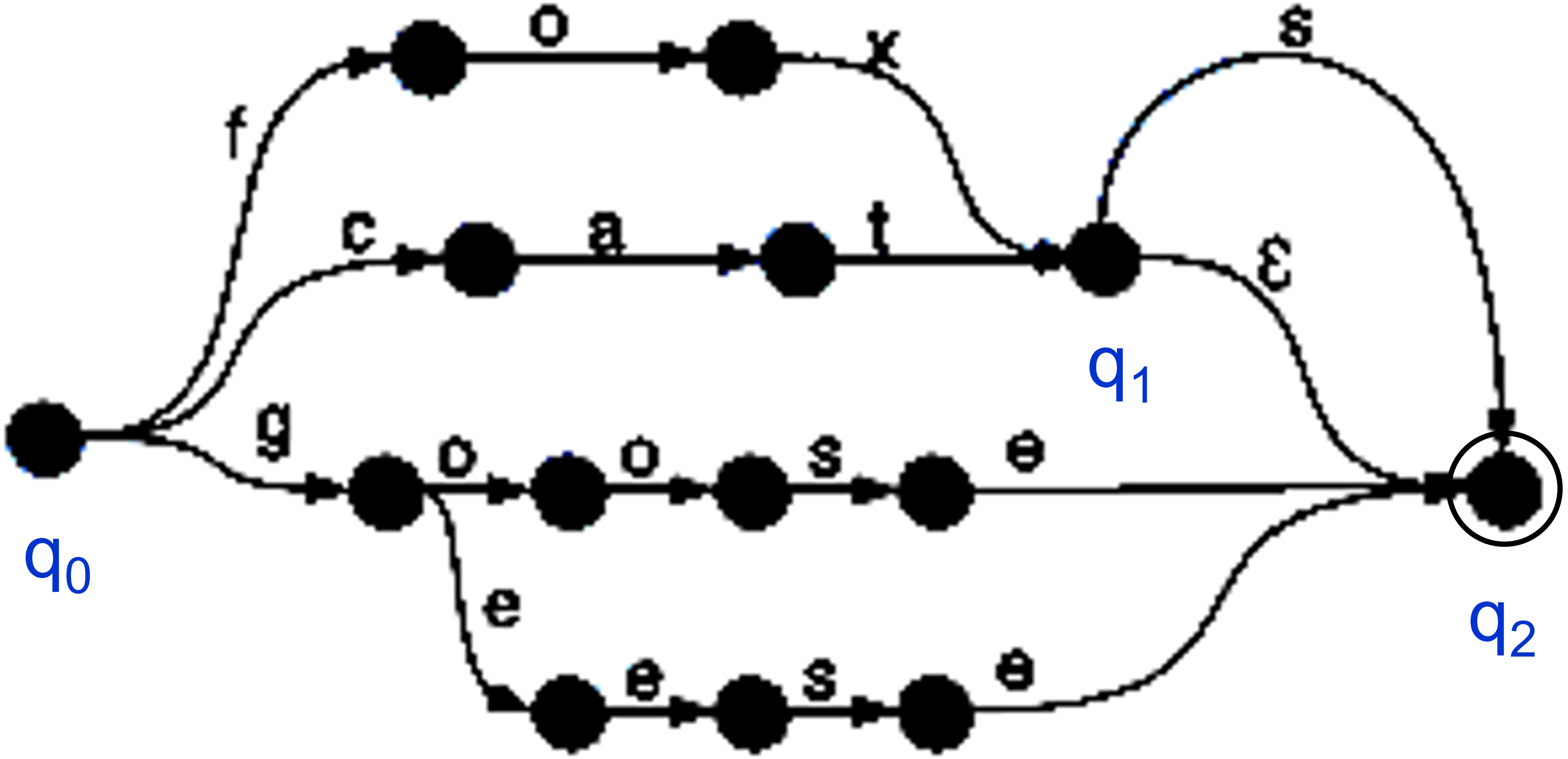


reg-noun	irreg-pl-noun	irreg-sg-noun	plural
fox	geese	goose	-s
cat	sheep	sheep	
aardvark	mice	mouse	

An acceptor

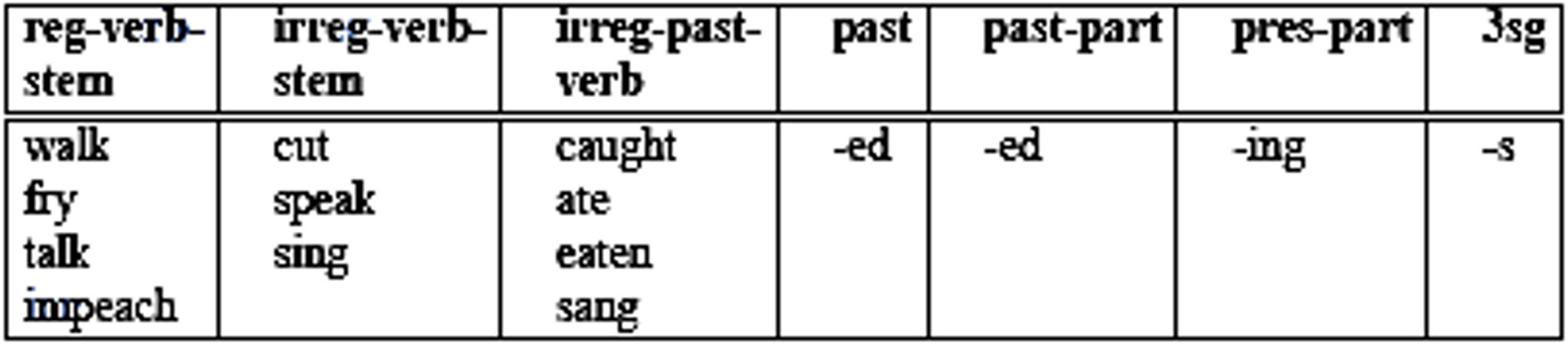


Expanded FSA



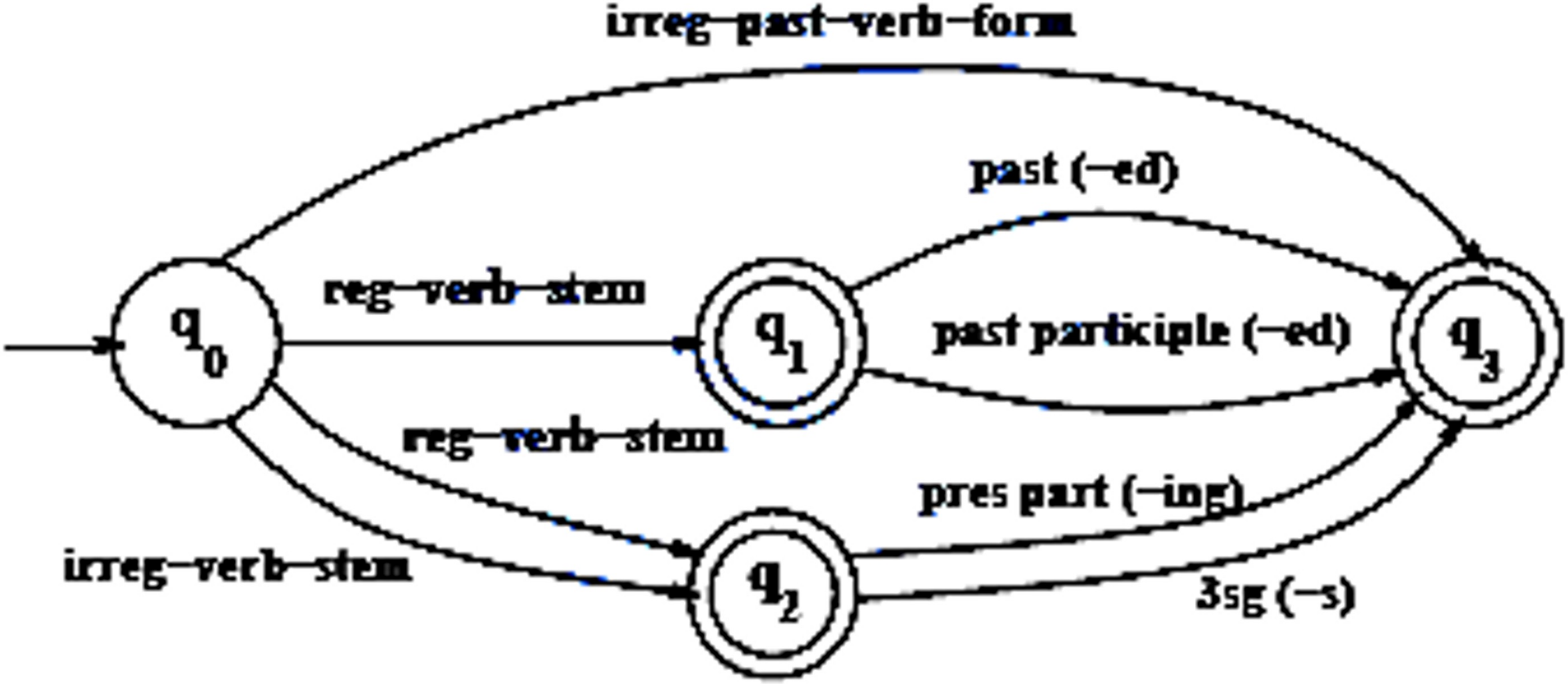
Lexicon for English verbs

- fly: v, +base, +irreg ⬄ irreg-verb-stem
- flew: v, +past, +irreg ⬄ irreg-past-verb
- walk: v, +base, +reg ⬄ reg-verb-stem

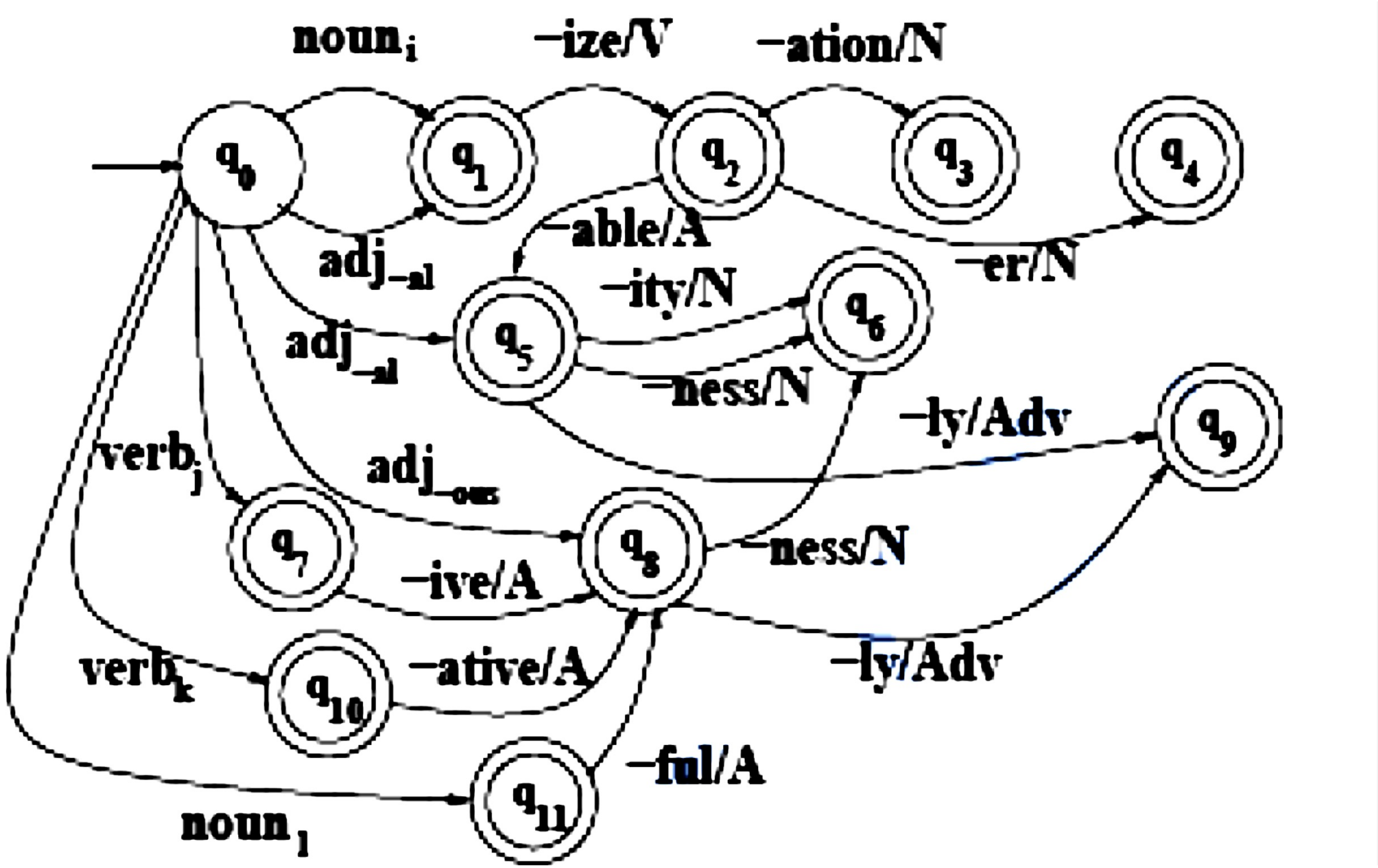


reg-verb-stem	irreg-verb-stem	irreg-past-verb	past	past-part	pres-part	3sg
walk	cut	caught	-ed	-ed	-ing	-s
fry	speak	ate				
talk	sing	eaten				
impeach		sang				

An FSA for the English verb



An FSA for English derivational morphology



So far

- Ex: cats
  - Have the entry “cat: reg-noun” in the lexicon
  - A path: q_0 ➔ q_1 ➔ q_2
  - Result: cats ➔ cat s ➔ cat^s#

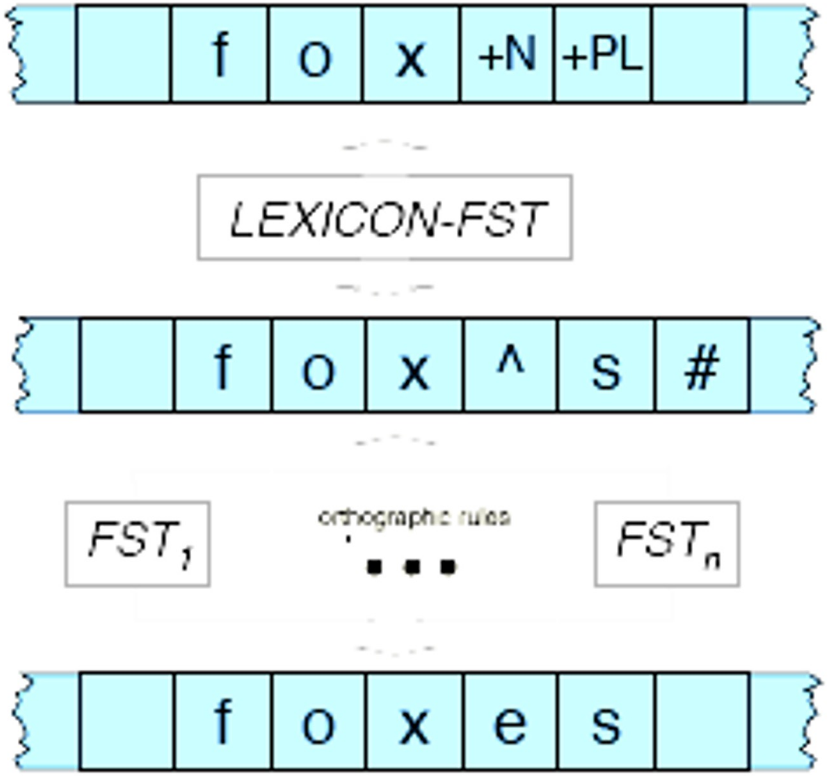
- Ex: civilize
  - Have the entry “civil: noun1” in the lexicon
  - A path: q_0 ➔ q_1 ➔ q_2
  - Result: civilize ➔ civil^ize#

- Remaining issues:
  - cat^s# ➔ cat +N +PL
  - spelling changes: foxes ➔ fox^s#

FST morphological analysis

- English morphology: J&M 3.1
- FSA acceptor: J&M 3.3
  - Ex: cats ➔ yes/no, foxs ➔ yes/no
- **FSTs for morphological analysis: J&M 3.5**
  - Ex: fox +N +PL ➔ fox^s#
- Adding orthographic rules: J&M 3.6-3.7
  - Ex: fox^s# ➔ foxes#

Three levels



Lexical level: f o x +N +PL

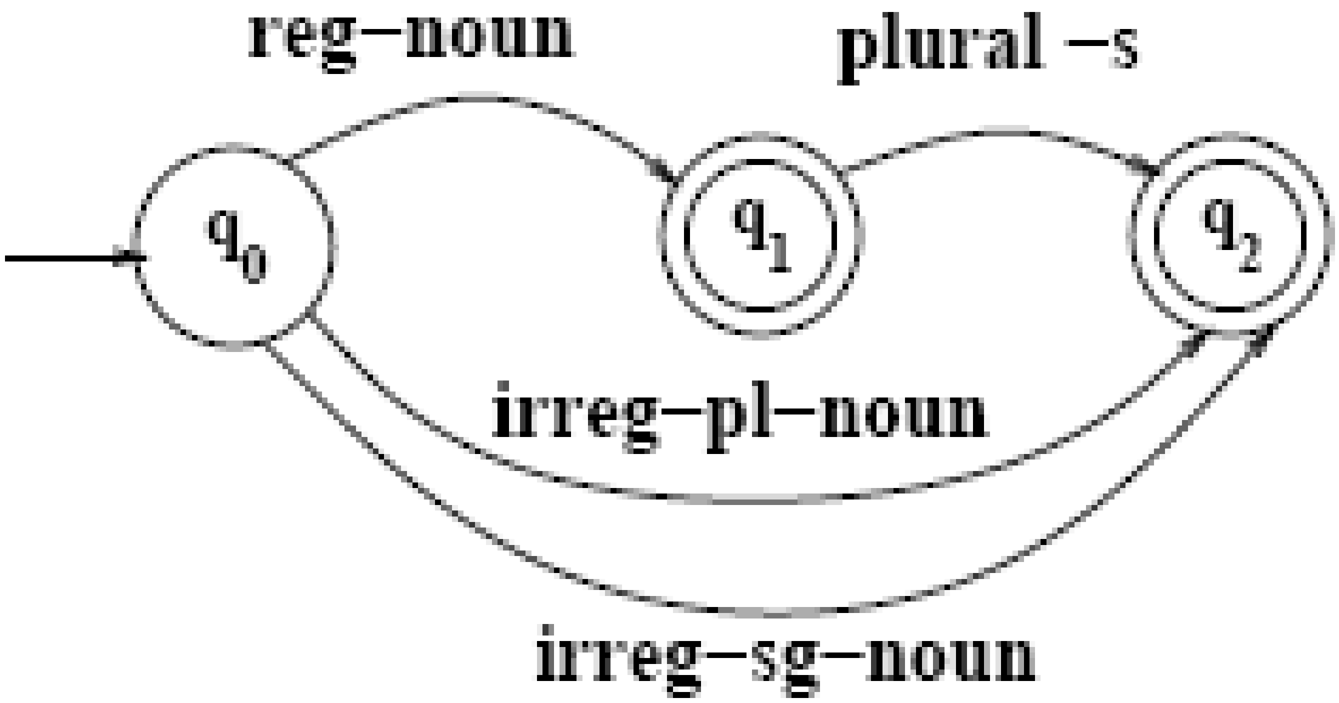
LEXICON-FST

Intermediate level: f o x ^ s #

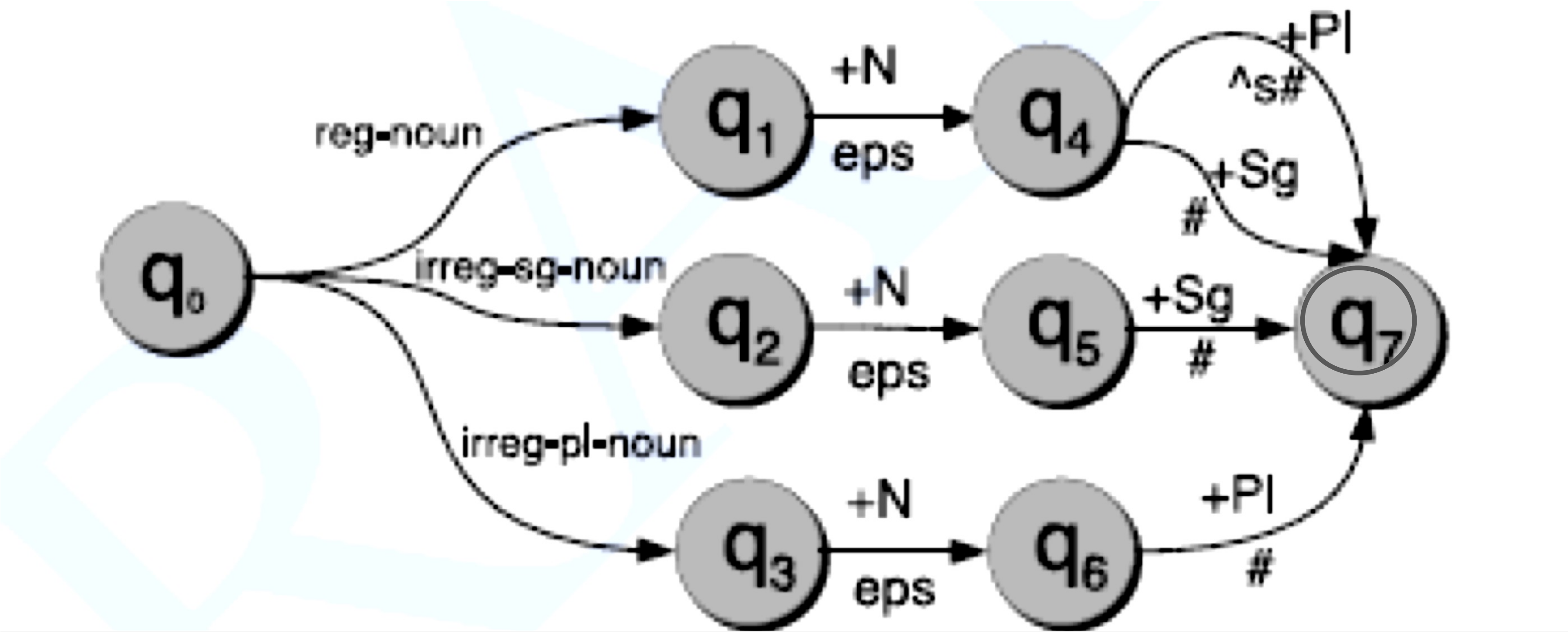
FST1 orthographic rules ... FSTn

Surface level: f o x e s

An acceptor



An FST



cat +N +PL ➔ cat^s#

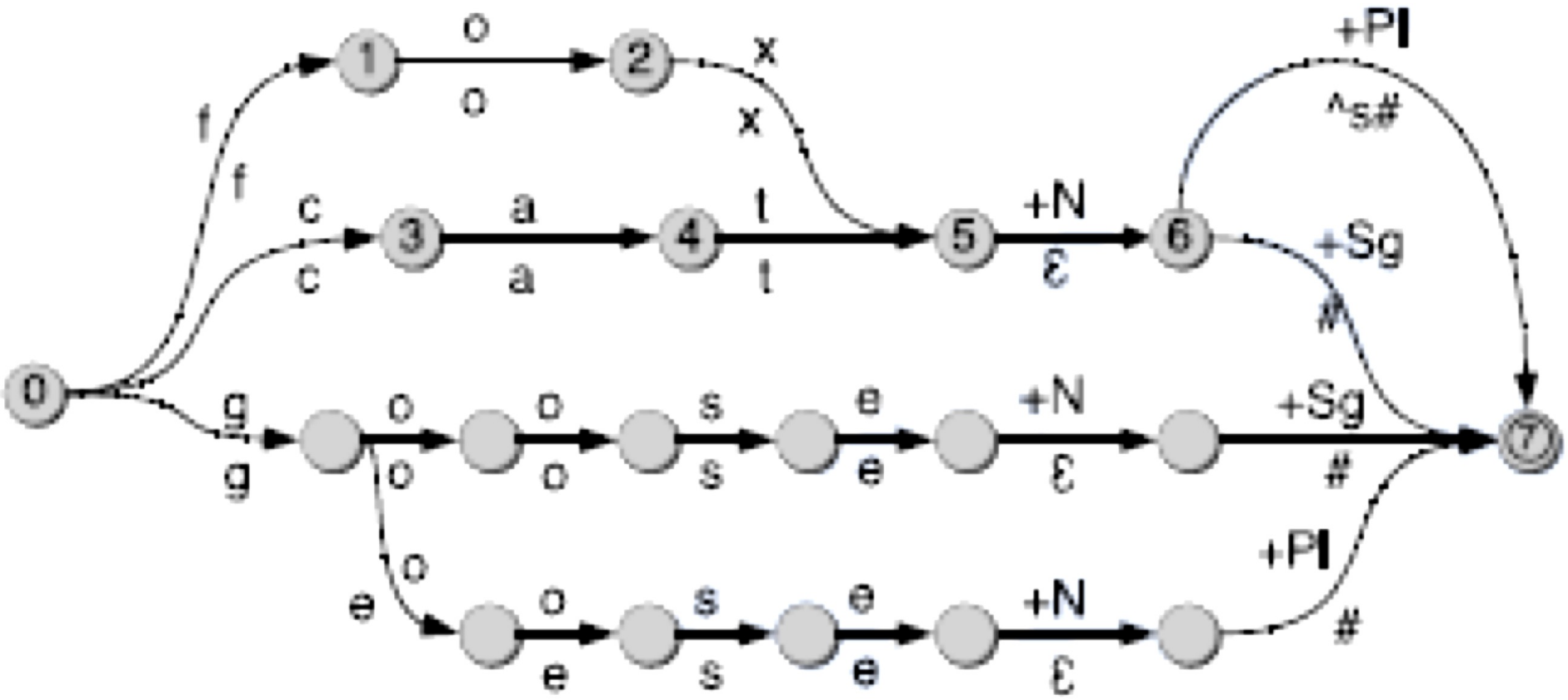
cat +N +Sg ➔ cat#

The lexicon for FST

reg-non	Irreg-pl-noun	Irreg-sg-noun
fox	g o:e o:e s e	goose
cat	sheep	sheep
aardvark	m o:i u:ϵ s:c e	mouse

goose → geese
mouse → mice

Expanding FST



fox +N + Pl ➔ fox^s#

cat +N +Pl ➔ cat^s#

goose +N +Sg ➔ goose#

goose +N +Pl ➔ geese#

FST morphological analysis

- English morphology: J&M 3.1
- FSA acceptor: J&M 3.3
  - Ex: cats ➔ yes/no, foxs ➔ yes/no
- FSTs for morphological analysis: J&M 3.5
  - Ex: fox +N +PL ➔ fox^s#
- **Adding orthographic rules: J&M 3.6-3.7**
  - Ex: fox^s# ➔ foxes#

Summary of FST morphological analyzer

- Three components:
  - Lexicon
  - Morphotactics
  - Orthographic rules

- Representing morphotactics as FST and expand it with the lexicon entries.

- Representing orthographic rules as FSTs.

- Combining all FSTs with operations such as composition.

- Giving the three components, creating and combining FSTs can be done automatically.

Remaining issues

- Creating the three components by hand is time consuming.

  ➔ unsupervised morphological induction

- How would a morphological analyzer help a particular application (e.g., IR, MT)?

How does the induction work?

- Start from a simple list of words and their frequencies:
  - Ex:

play	67
played	100
walked	40
walk	21

- Try to find the most efficient way to encode the wordlist:
  - Ex: minimum description length (MDL)

General approach

- Initialize: start from an initial set of “words” and find the description length of this set

- Repeat until convergence
  - Generate a candidate set of new “words” that will each enable a reduction in the description length

- Ex: walk, walked, play, played
  - four words
  - two words (walk and play) and a suffix (-ed)

Additional slides

Orthographic rules

- E insertion: fox ➔ foxes
- 1st try: ϵ ➔ e

- “e” is added after -s, -x, -z, etc. before -s
- 2nd try: ϵ ➔ e / (s|x|z|) _ s
- Problem?
  - Ex: glass ➔ glases

- 3rd try: ϵ ➔ e / (s|x|z)^_ s#

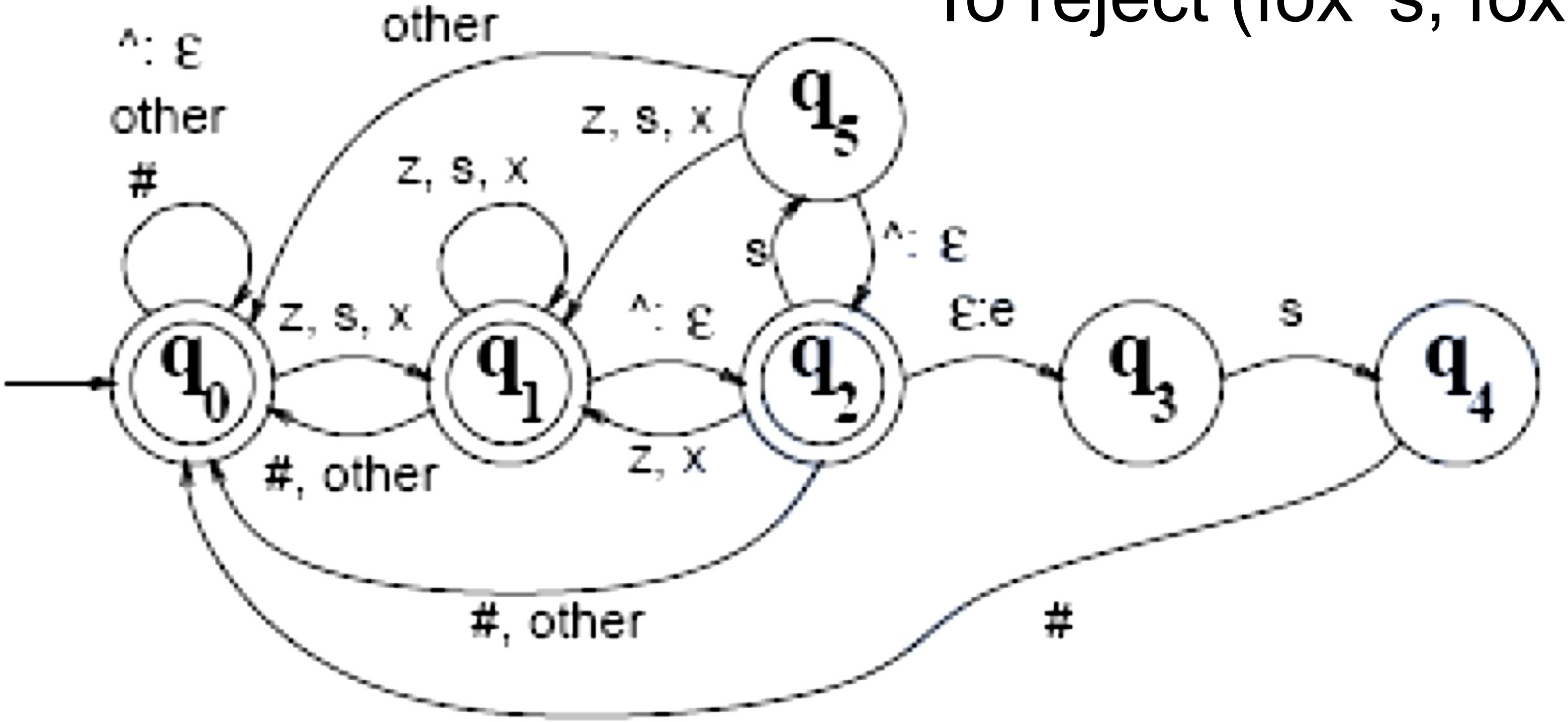
Rewrite rules

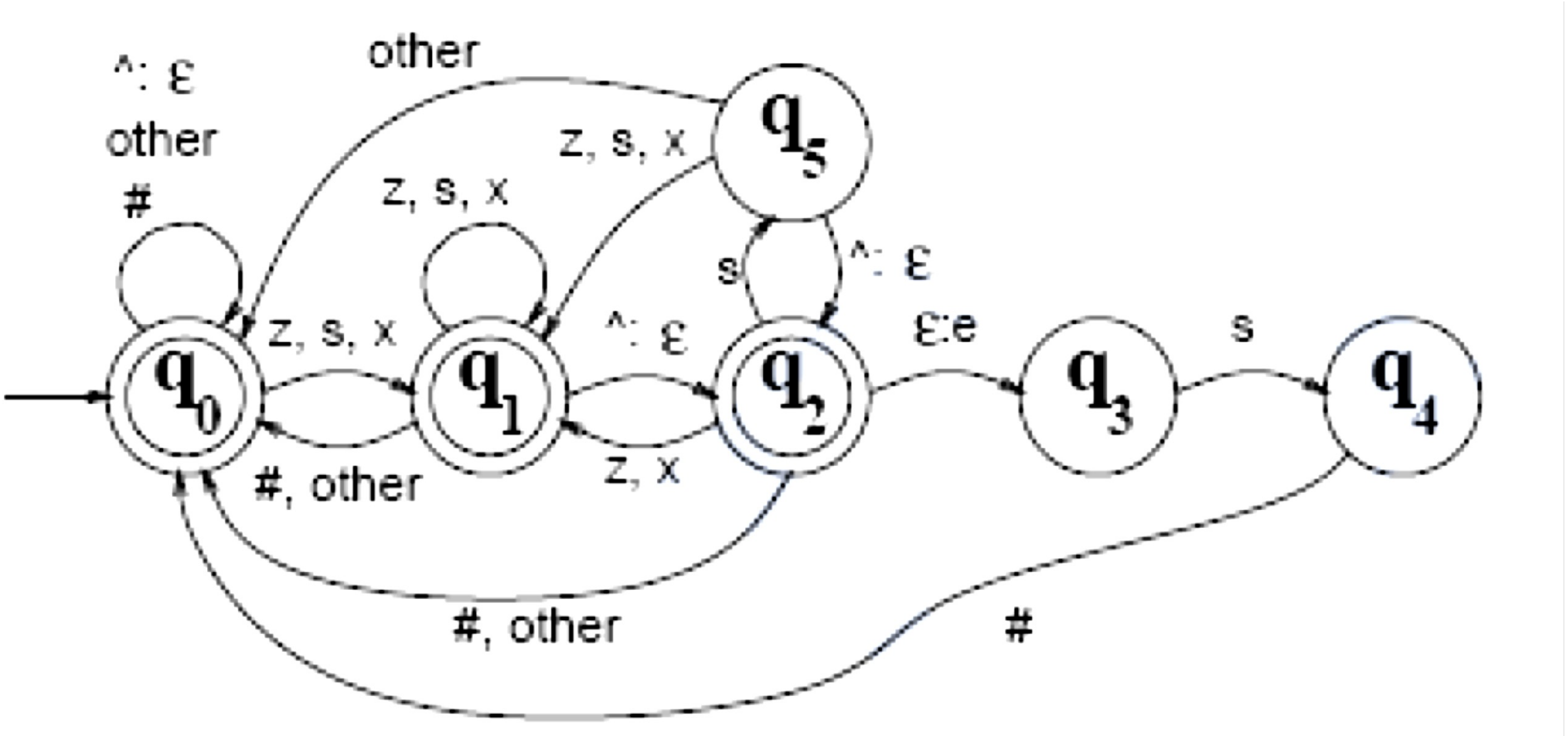
- Format: $\alpha \longrightarrow \beta / \lambda \_ \rho$
- Rewrite rules can be optional or obligatory
- Rewrite rules can be ordered to reduce ambiguity.
- Under some conditions, these rewrite rules are equivalent to FSTs.
  - α is not allowed to match something introduced in the previous rule application

Representing orthographic rules as FSTs (**)

- ϵ ➔ e / (z|s|x)^_ s#
- Input: …(z|s|x)^s# immediate level
- Output: …(z|s|x)es# surface level

To reject (fox^s, foxs)





(fox, fox): q0, q0, q0, q1, acc

(fox#, fox#): q0, q0, q0, q1, q0, acc

(fox^z#, foxz#), q0, q0, q0, q1, q2, q1, q0, acc

(fox^s#, foxes#): q0, q0, q0, q1, q2, q3, q4, q0, acc

(fox^s, foxs): q0, q0, q0, q1, q2, q5 reject

What would the FST accept?

(f, f)

(fox, fox)

(fox#, fox#)

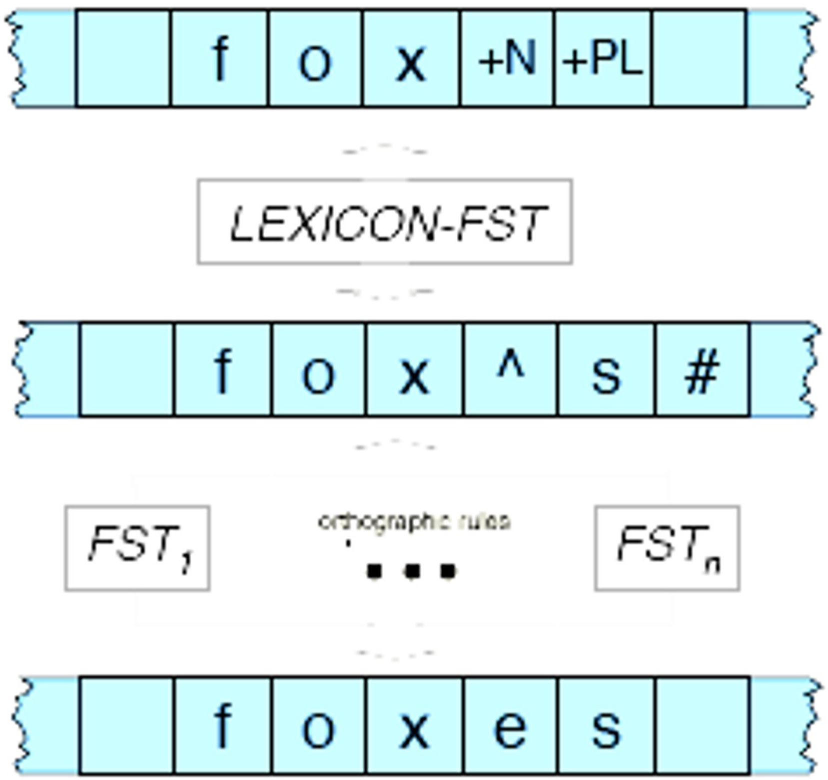
(fox^z#, foxz#)

(fox^s#, foxes#)

It will reject:

(fox^s, foxs)

Combining lexicon and rules



Lexical level: f o x +N +PL

LEXICON-FST

Intermediate level: f o x ^ s #

FST_1 orthographic rules ... FST_n

Surface level: f o x e s