Words:
Surface Variation and Automata

CMSC 35100
Natural Language Processing
April 3, 2003

Roadmap

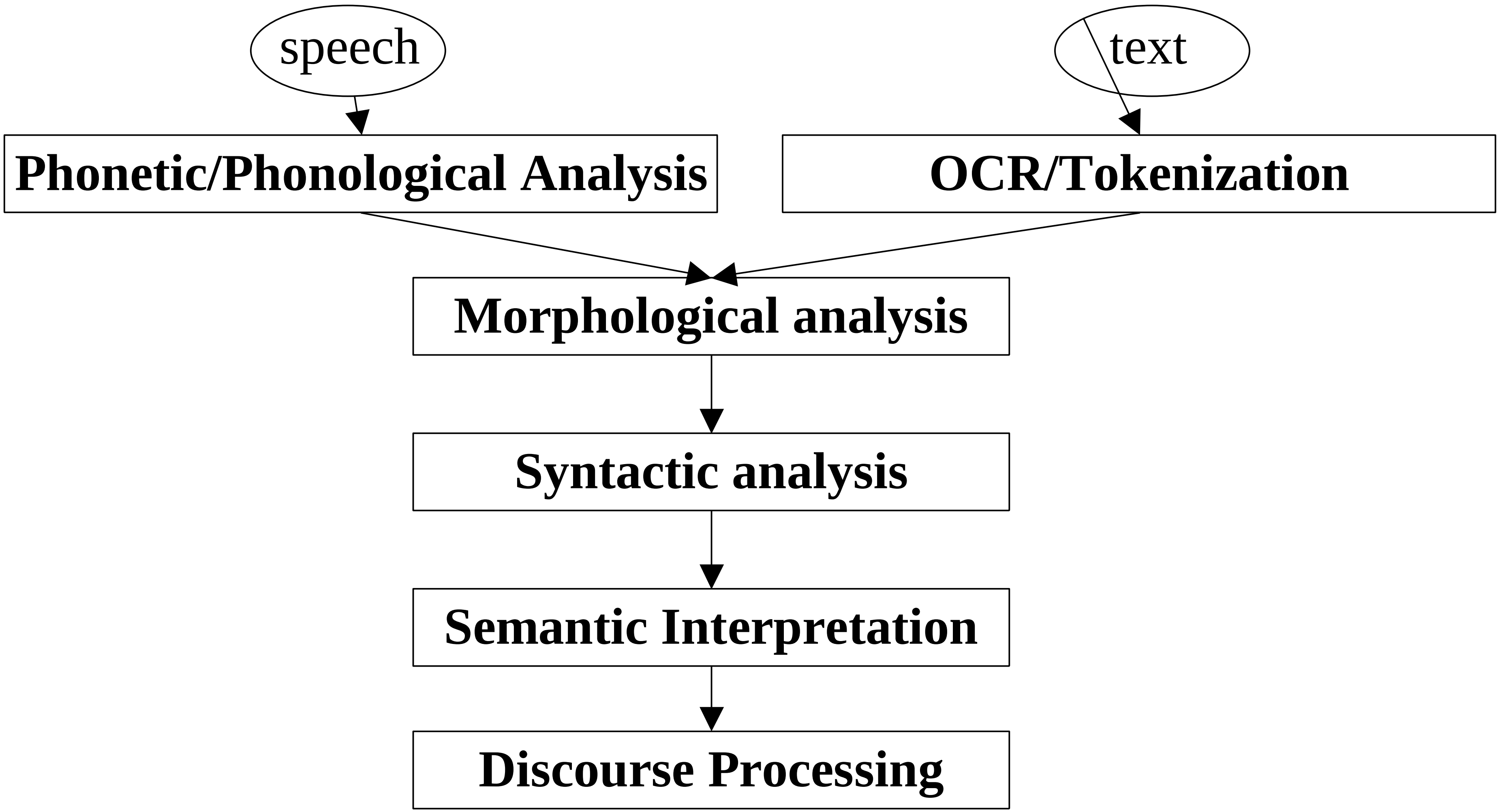
- The NLP Pipeline
- Words: Surface variation and automata
  - Motivation:
    - Morphological and pronunciation variation
  - Mechanisms:
    - Patterns: Regular expressions
    - Finite State Automata and Regular Languages
      - Non-determinism, Transduction, and Weighting
  - FSTs and Morphological/Phonological Rules

Real Language Understanding

- Requires more than just pattern matching
- But what?,

- 2001:
- Dave: Open the pod bay doors, HAL.
- HAL: I'm sorry, Dave. I'm afraid I can't do that.

Language Processing Pipeline



Phonetics and Phonology

- Convert an acoustic sequence to word sequence
- Need to know:
  - Phonemes: Sound inventory for a language
  - Vocabulary: Word inventory – pronunciations
  - Pronunciation variation:
    - Colloquial, fast, slow, accented, context

Morphology & Syntax

- Morphology: Recognize and produce variations in word forms
  - (E.g.) Inflectional morphology:
    - e.g. Singular vs plural; verb person/tense
      - Door + sg: door
      - Door + plural: doors
      - Be + 1st person, sg, present: am
- Syntax: Order and group words together in sentence
      - Open the pod bay doors
      - Vs
      - Pod the open doors bay

Semantics

- Understand word meanings and combine meanings in larger units

- Lexical semantics:
  - Bay: partially enclosed body of water; storage area
- Compositional sematics:
  - “pod bay doors”:
    - Doors allowing access to bay where pods are kept

Discourse & Pragmatics

- Interpret utterances in context
  - Resolve references:
    - “I'm afraid I can't do that”
      - “that” = “open the pod bay doors”
  - Speech act interpretation:
    - “Open the pod bay doors”
      - Command

Surface Variation: Morphology

- Searching for documents about
  - “Televised sports”
- Many possible surface forms:
  - Televised, televise, television, ..
  - Sports, sport, sporting
- Convert to some common base form
  - Match all variations
  - Compact representation of language

Surface Variation: Morphology

- Inflectional morphology:
  - Verb: past, present; Noun: singular, plural
  - e.g. Televise: inf; televise +past -> televised
  - Sport+sg: sport; sport+pl: sports
- Derivational morphology:
  - v->n: televise -> television
- Lexicon:Root form + morphological features
- Surface: Apply rules for combination
- Identify patterns of transformation, roots, affixes

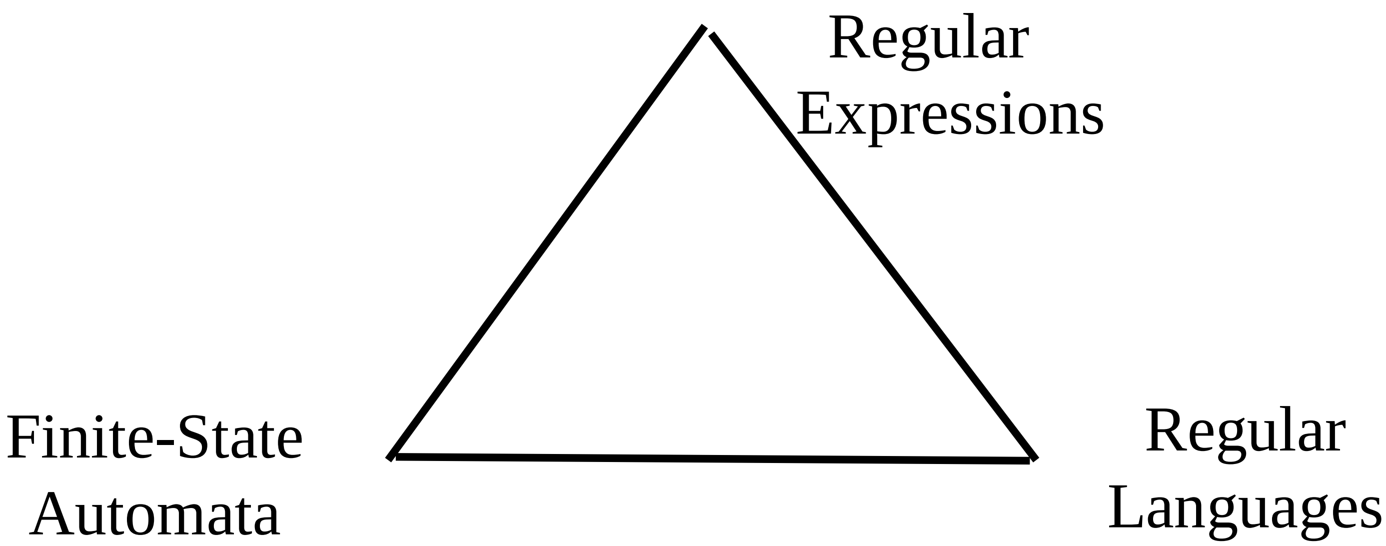
Surface Variation: Pronunciation

- Regular English plural: +s
- English plural pronunciation:
  - cat+s -> cats where s=*s, but*
  - *dog+s* -> *dogs where s=z, and*
  - *base+s* -> *bases where s=iz*
- Phonological rules govern morpheme combination
  - *+s = s, unless [voiced]+s = z, [sibilant]+s= iz*
- Common lexical representation
  - Mechanism to convert appropriate surface form

Representing Patterns

- Regular Expressions
  - Strings of 'letters' from an alphabet Sigma
  - Combined by concatenation, union, disjunction, and Kleene *
- Examples: a, aa, aabb, abab, baaa!, baaaaaa!
  - Concatenation: ab
  - Disjunction: a[abcd]: -> aa, ab, ac, ad
    - With precedence: gupp(y|ies) -> guppy, guppies
  - Kleene : (0 or more): baa*! -> ba!, baa!, baaaaa!

Expressions, Languages & Automata

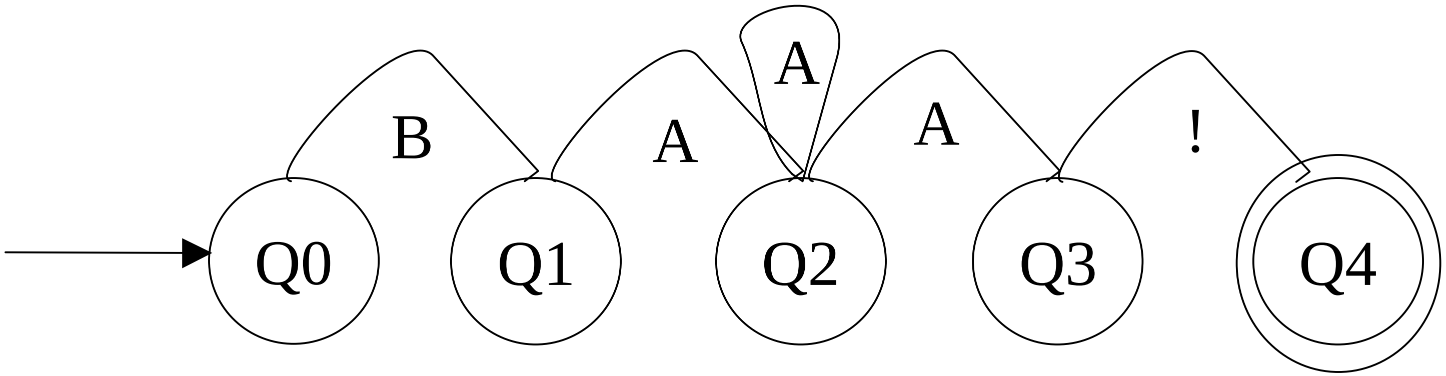


- Regular expressions specify sets of strings (languages) that can be implemented with a finite-state automaton.

Finite-State Automata

- Formally,
  - Q: a finite set of N states: *q0*, *q1*,...,*qN*
    - Designated start state: q0; final states: F
  - Sigma: alphabet of symbols
  - Delta(q,i): Transition matrix specifies in state q, on input i, the next state(s)
- Accepts a string if in final state at end of string
  - O.W. Rejects

Finite-State Automata



- Regular Expression: baaa*!
  - e.g. Baaaa!
- Closed under concatention, union, disjunction, and Kleene *

Non-determinism & Search

- Non-determinism:
  - Same state, same input -> multiple next states
  - E.g.: Delta(q2,a)-> q2, q3
- To recognize a string, follow state sequence
  - Question: which one?
  - Answer: Either!
    - Provide mechanism to backup to choice point
      - Save on stack: LIFO: Depth-first search
      - Save in queue: FIFO: Breadth-first search
- NFSA equivalent to FSA
  - Requires up to 2^n states, though

From Recognition to Transformation

- FSAs accept or reject strings as elements of a regular language: recognition
- Would like to extend:
  - Parsing: Take input and produce structure for it
  - Generation: Take structure and produce output form
  - E.g. Morphological parsing: words -> morphemes
    - Contrast to stemming
  - E.g. TTS: spelling/representation -> pronunciation

Morphology

- Study of minimal meaning units of language
  - Morphemes
    - Stems: main units; Affixes: additional units
    - E.g. Cats: stem=cat; affix=s (plural)
  - Inflectional vs Derivational:
    - Inflection: add morpheme, same part of speech
      - E.g. Plural -s of noun; -ed: past tense of verb
    - Derivation: add morpheme, change part of speech
      - E.g. verb+ation -> noun; realize -> realization
- Huge language variation:
    - English: relatively little: concatenative
    - Arabic: richer, templatic kCtCb + -s: kutub
    - Turkish: long affix strings, “agglutinative”

Morphology Issues

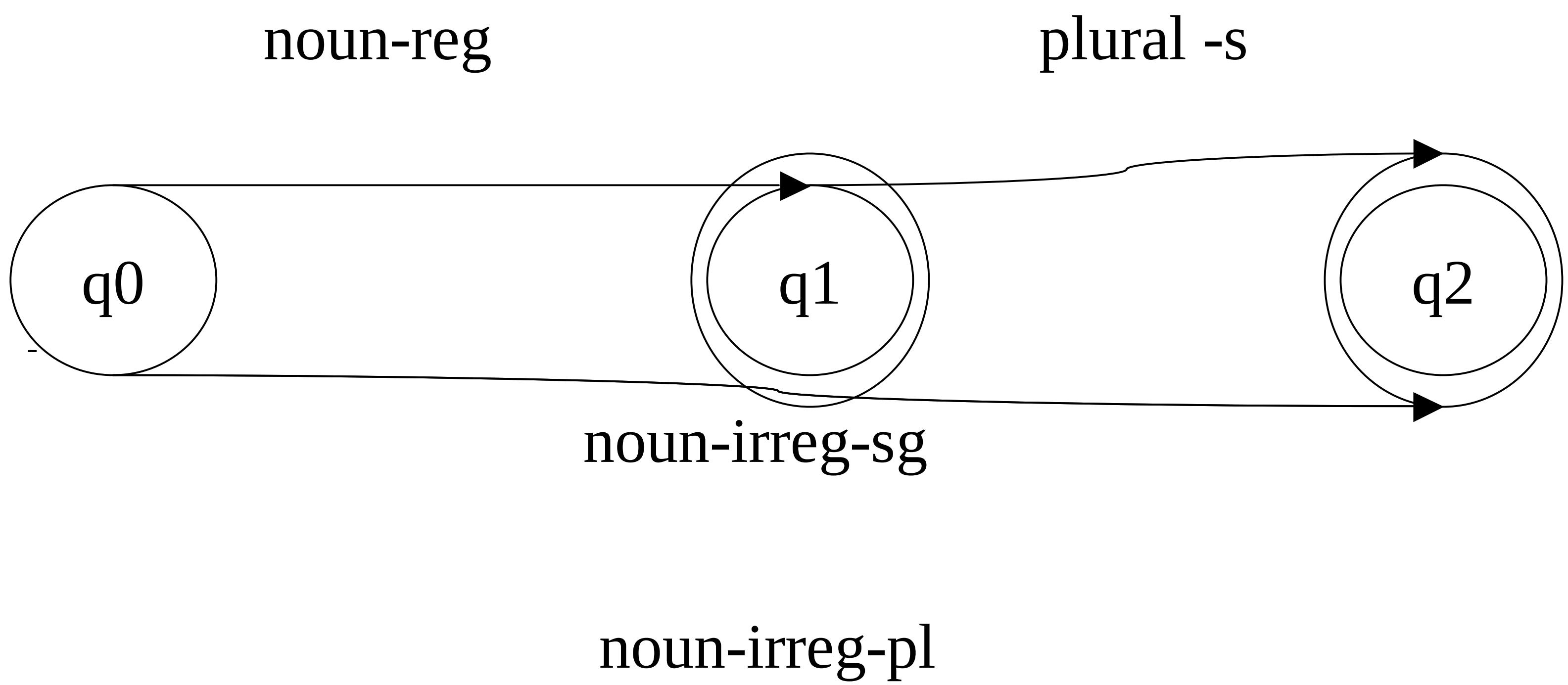
- Question 1: Which affixes go with which stems?
  - Tied to POS (e.g. Possessive with noun; tenses: verb)
  - Regular vs irregular cases
    - Regular: majority, productive – new words inherit
    - Irregular: small (closed) class – often very common words

- Question 2: How does the spelling change with the affix?
  - E.g. Run + ing -> running; fury+s -> furies

Associating Stems and Affixes

- Lexicon
  - Simple idea: list of words in a language
  - Too simple!
    - Potentially HUGE: e.g. Agglutinative languages
  - Better:
    - List of stems, affixes, and representation of morphotactics
    - Split stems into equivalence classes w.r.t. morphology
      - E.g. Regular nouns (reg-noun) vs irregular-sg-noun...
- FSA could accept legal words of language
  - Inputs: words-classes, affixes

Automaton for English Nouns



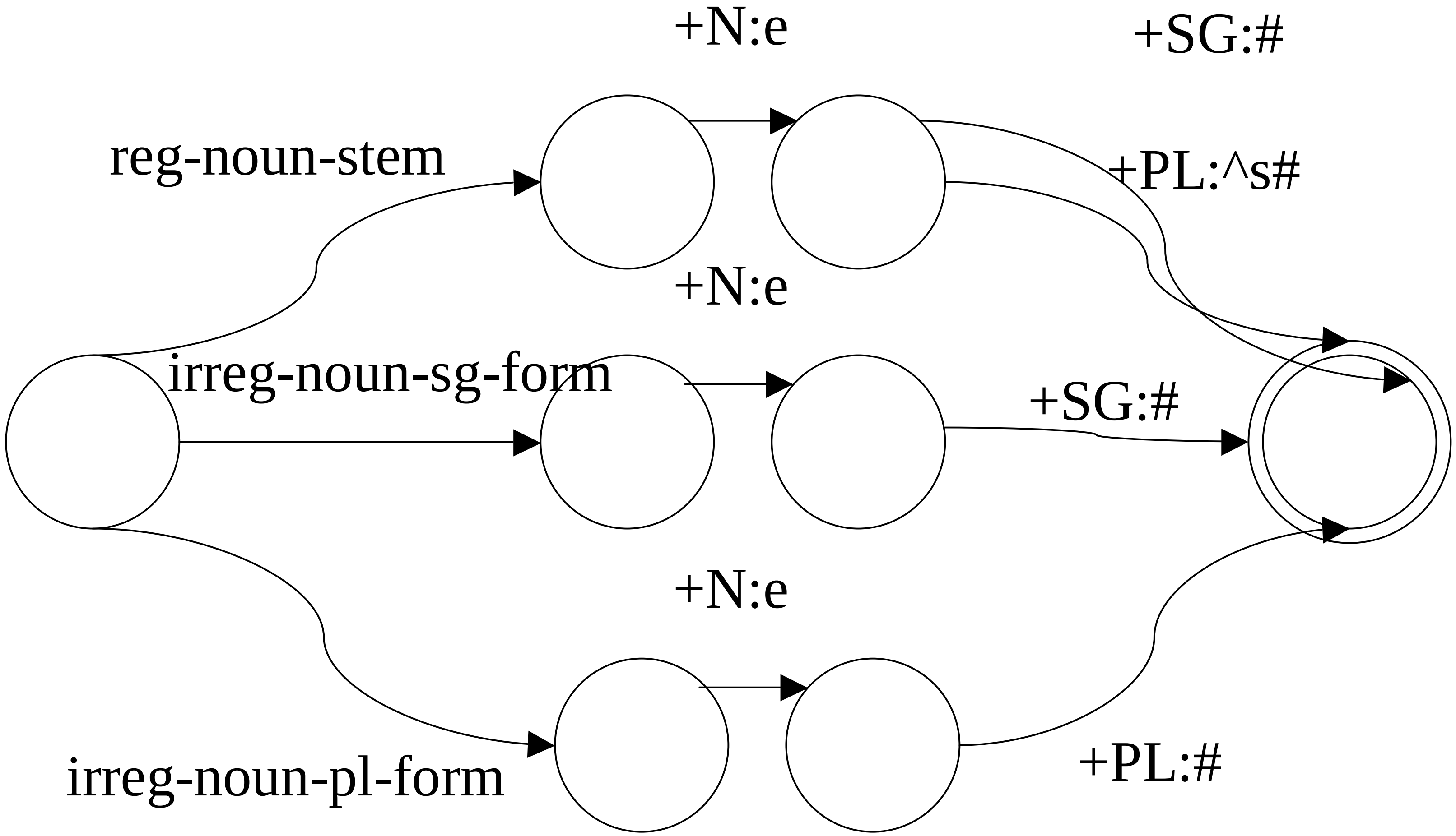
Two-level Morphology

- Morphological parsing:
  - Two levels: (Koskenniemi 1983)
    - Lexical level: concatenation of morphemes in word
    - Surface level: spelling of word surface form
  - Build rules mapping between surface and lexical
- Mechanism: Finite-state transducer (FST)
  - Model: two tape automaton
  - Recognize/Generate pairs of strings

FSA -> FST

- Main change: Alphabet
  - Complex alphabet of pairs: input x output symbols
  - e.g. i:o
    - Where i is in input alphabet, o in output alphabet
- Entails change to state transition function
  - Delta(q, i:o): now reads from complex alphabet
- Closed under union, inversion, and composition
  - Inversion allows parser-as-generator
  - Composition allows series operation

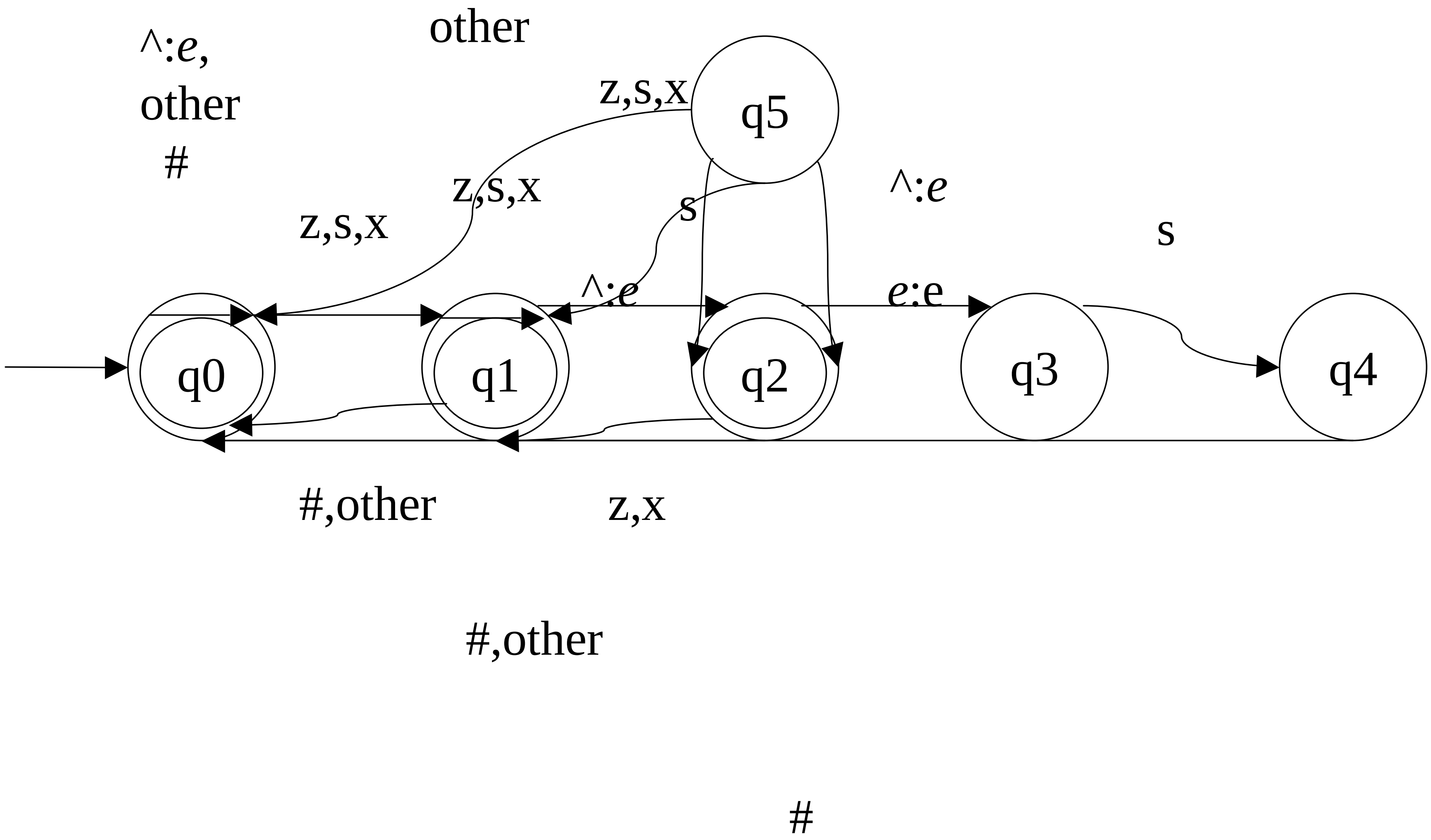
Simple FST for Plural Nouns



Rules and Spelling Change

- Example: E insertion in plurals
  - After x, z, s...: fox + -s -> foxes
- View as two-step process
  - Lexical -> Intermediate (create morphemes)
  - Intermediate -> Surface (fix spelling)
- Rules: (a la Chomsky & Halle 1968)
  - Epsilon -> e/{x,z,s}^__s#
    - Rewrite epsilon (empty) as e when it occurs between x,s,or z at end of one morpheme and next morpheme is -s

E-insertion FST



Implementing Parsing/Generation

- Two-layer cascade of transducers (series)
  - Lexical -> Intermediate; Intermediate -> Surface
    - I->S: all the different spelling rules in parallel
- Bidirectional, but
  - Parsing more complex
    - Ambiguous!
      - E.g. Is fox noun or verb?

Shallow Morphological Analysis

- Motivation: Information Retrieval
  - Just enable matching – without full analysis
- Stemming:
  - Affix removal
    - Often without lexicon
    - Just return stems – not structure
  - Classic example: Porter stemmer
    - Rule-based cascade of repeated suffix removal
      - Pattern-based
    - Produces: non-words, errors, ...

Automatic Acquisition of Morphology

- “Statistical Stemming” (Cabezas, Levow, Oard)
  - Identify high frequency short affix strings for removal
  - Fairly effective for Germanic, Romance languages
- Light Stemming (Arabic)
  - Frequency-based identification of templates & affixes
- Minimum description length approach
    - (Brent and Cartwright1996, DeMarcken 1996, Goldsmith 2000
  - Minimize cost of model + cost of lexicon | model
-