Finite State Machines

Lecture #8

Introduction to Natural Language Processing

CMPSCI 585, Spring 2004

University of Massachusetts Amherst

Charles Sutton

Overview

Applications

- Language modeling
- Morphology
- Word segmentation

Techniques

- Finite State Automata (FSAs)
- Finite State Transducers (FSTs)
- Weighted FSAs/FSTs

Finite State Automata

FSAs (formally)

Regular Languages

Is English regular?
(center embedded sentences)

The cat likes tuna fish.

The cat the dog chased likes tuna fish.

The cat the dog the rat bit chased likes tuna fish.

The cat the dog the rat the elephant admired bit chased likes tuna fish.

Morphology

Morphology: The study of the internal structure of words

Words can be divided into **morphemes,** e.g.,

walked: **walk** + ed

happier: **happy** + er

antidisestablishmentarianism:

anti + dis + **establish** + ment + arian + ism

Inflectional Morphology

Inflection: Combination of stem with morphemes, usually for syntactic function like agreement

woodchuck	→	woodchucks
fox	→	foxes
cherub	→	cherubim
walk	→	walked
cut	→	cutting
catch	→	caught

Derivational Morphology

Derivation: Combination of stem and other morpheme results in usually different class, usually with different meaning

embrace → embraceable

able → unable

slow → slowly

big ↛ *unbigly

colony → colonization

Lexicon as Big List

Could store lexicon as a Big Fat List (BFL), but:

- Takes a lot of space
- Regular inflection is **productive**: e.g., What's the plural of Segway?
- Regularities between affixes (**morphotactics**), e.g., +ation occurs after +ize

FSA Lexicon

Problems with this FSA…

Works fine as a way of storing a lexicon

Doesn't tell us how to inflect new words

(e.g., fax inflects just like fox)

Orthography

Spelling often changes as affixes are added

A	B	C
fox	fox+s	foxes
big	big+er	bigger
picnic	picnic+ing	picnicking
try	try+s	tries

Morphological Parsing

Use a finite-state machine to convert between

Lexical form	Surface form
fox+N+PL	foxes
big+ADJ+CMPR	bigger
picnic+V+PRES-PART	picnicking
try+V+PL	tries

Finite State Transducers

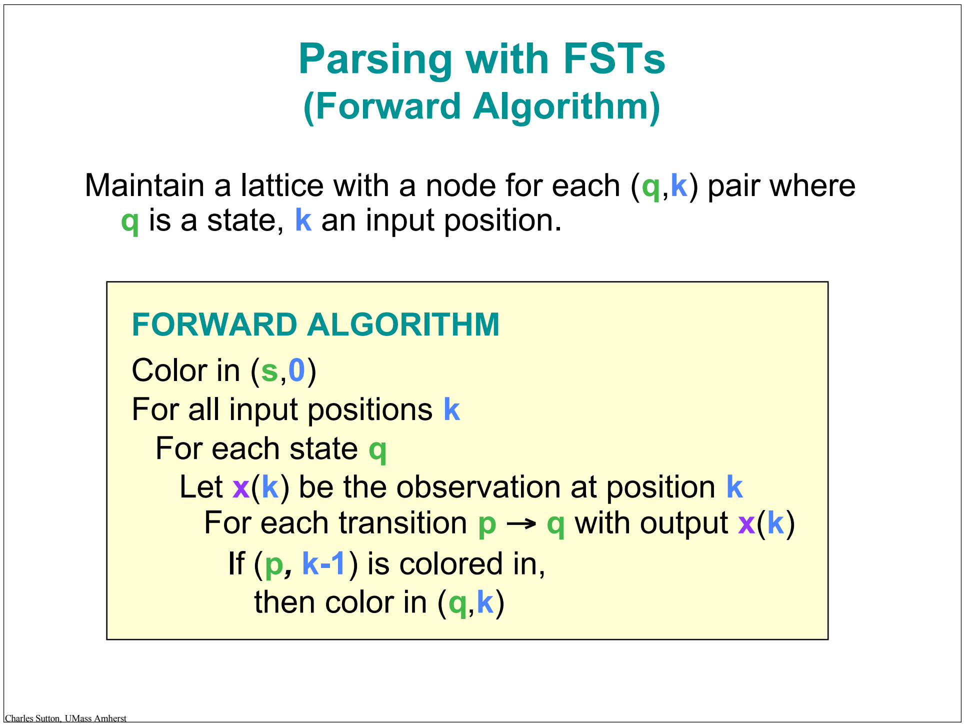
FSTs (formally)

Views of FSTs

- **Generator**: Given "meow", output "baaa"
- **Recognizer**: Accept/reject a language of pairs like "(meow, baaa)"
- **Parser**: What words of Cat could have produced "baaa"?

Parsing with FSTs

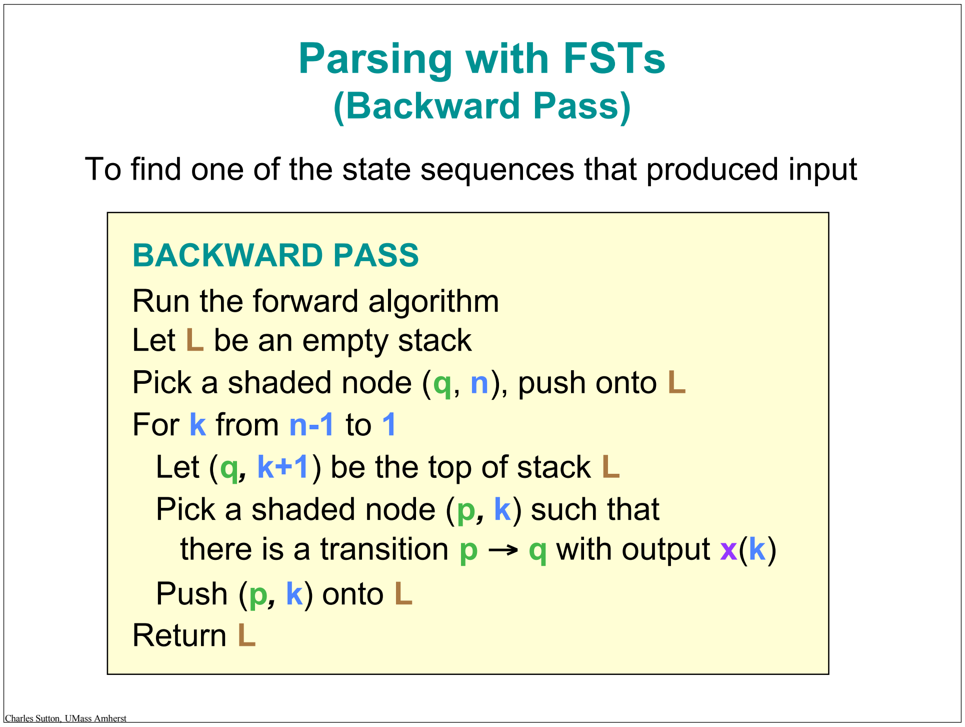
- Can convert into an inverse transducer by reversing input/output
- Result could be **nondeterministic**, i.e., several arcs for the same input
- Convert to deterministic FSTs (this is always possible)



Parsing with FSTs
(Forward Algorithm)

Maintain a lattice with a node for each (q,k) pair where q is a state, k an input position.

```
FORWARD ALGORITHM
Color in (s,0)
For all input positions k
  For each state q
    Let x(k) be the observation at position k
      For each transition p → q with output x(k)
        If (p, k-1) is colored in,
          then color in (q,k)
```

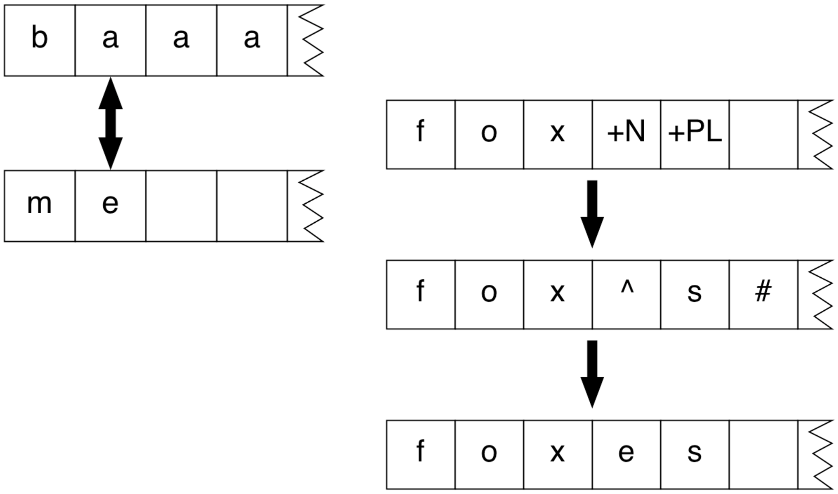


Parsing with FSTs
(Backward Pass)

To find one of the state sequences that produced input

```
BACKWARD PASS
Run the forward algorithm
Let L be an empty stack
Pick a shaded node (q, n), push onto L
For k from n-1 to 1
  Let (q, k+1) be the top of stack L
  Pick a shaded node (p, k) such that
    there is a transition p → q with output x(k)
  Push (p, k) onto L
Return L
```

FSTs for Morphology



b a a a

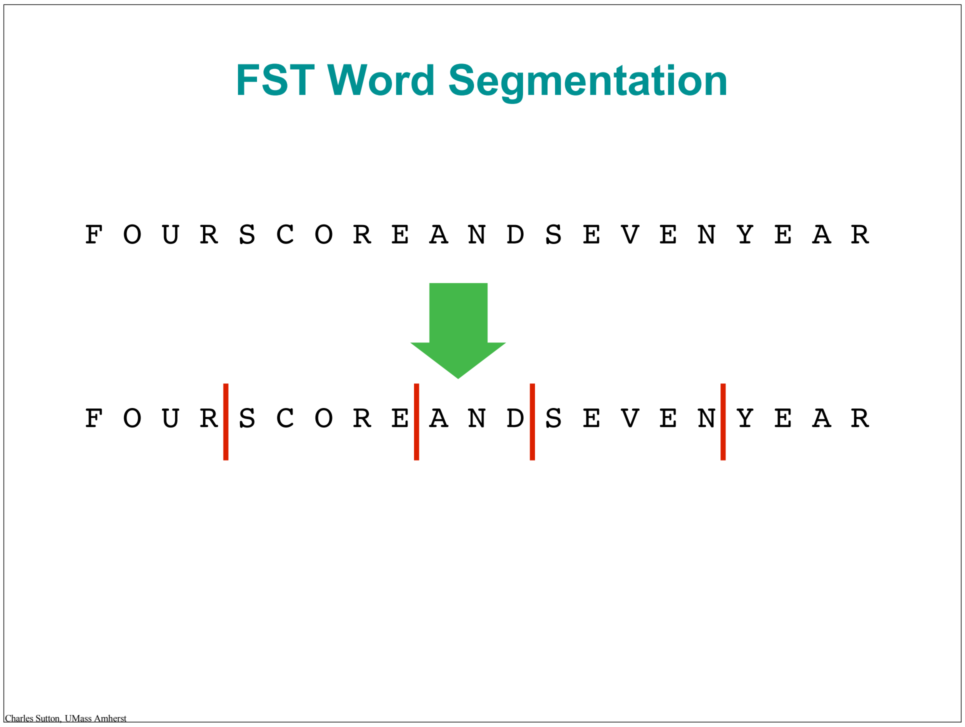
m e

f o x +N +PL

f o x ^ s #

f o x e s

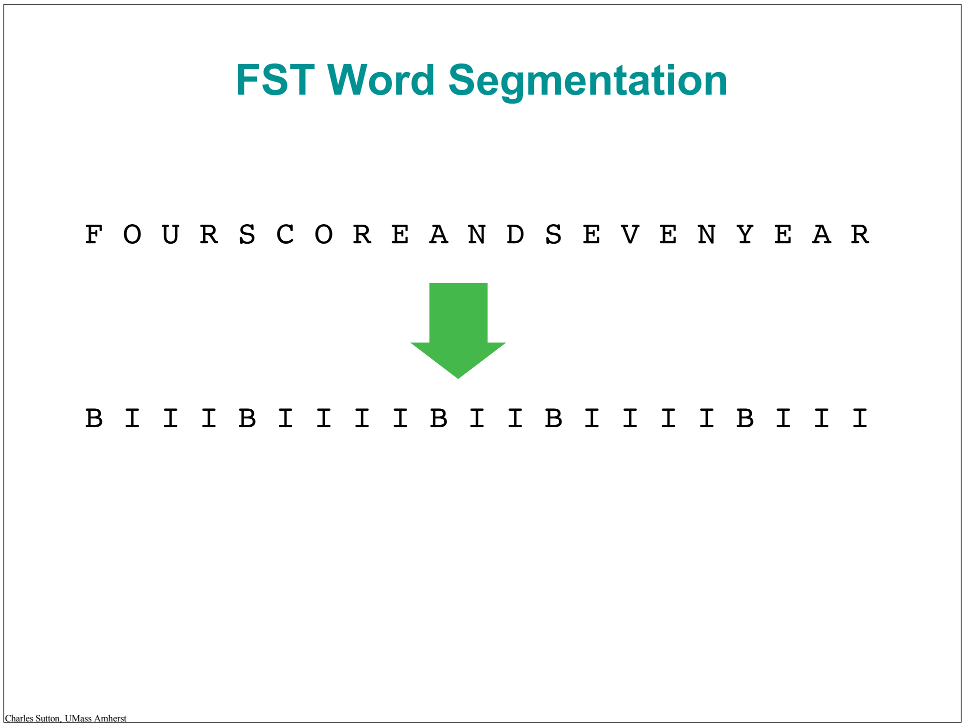
FSTs for Morphology



FST Word Segmentation

F O U R S C O R E A N D S E V E N Y E A R

F O U R | S C O R E | A N D | S E V E N | Y E A R

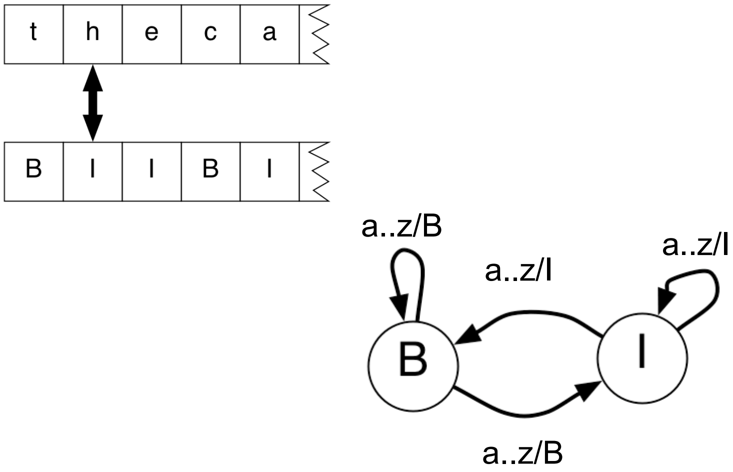


FST Word Segmentation

F O U R S C O R E A N D S E V E N Y E A R

B I I I B I I I I B I I B I I I I B I I I

FST Word Segmentation



t	h	e	c	a
B	I	I	B	I

Ambiguity

Weighted FSAs

Weighted FSTs