Foundations of Natural Language Processing
Lecture 11c
Finite State Transducers for Morphology

Alex Lascarides
(slides based on those of Shay Cohen)

School of informatics

Recap: FSTs for Morphological Parsing

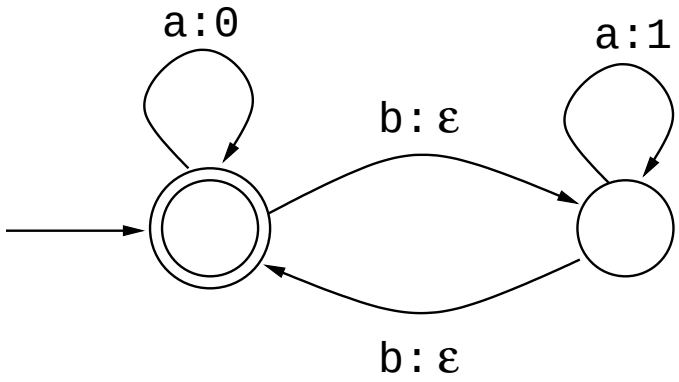
- Morphological parsing is the problem of extracting the lexical form from the surface form

Surface form:	walks	walking	geese
Intermediate form:	walk^ s#	walk^ ing#	geese^ #
Lexical form:	walk+N+PL walk+V+3SG	walk+V+PresPart	goose+N+PL

- We should take account of:
  - Irregular forms (e.g. goose → geese)
  - Systematic rules (e.g. 'e' inserted before suffix 's' after s,x,z,ch,sh: fox → foxes, watch → watches)
  - Things that look like affixes but aren't (*proactive* vs. *protect*)

- Finite State Transducers are a suitable way to capture valid mappings among surface, intermediate and lexical forms.

Simple (abstract) FST Example



Example: $aabaaabbbab \mapsto$ 001111

Today: Example Finite State Transducers for morphological analysis.

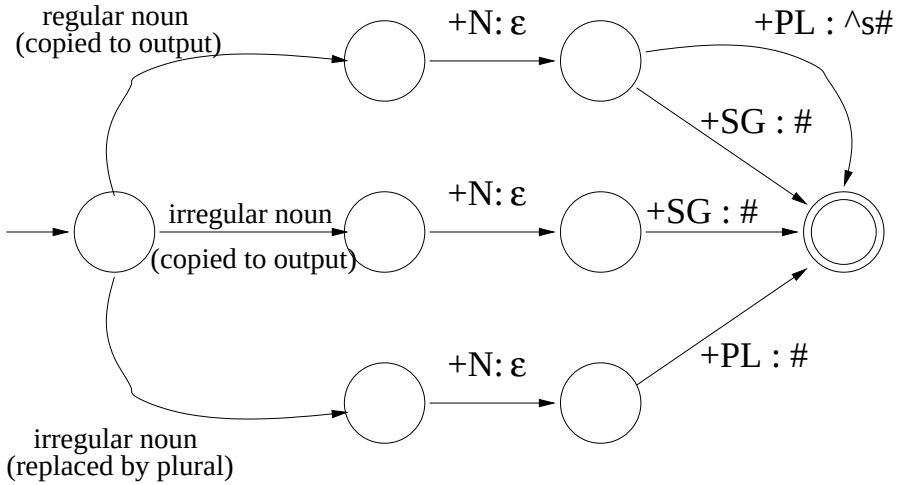
Formal definition for FSTs

- A finite state transducer T with inputs from Σ and outputs from Π consists of:
  - states Q, S (for start), F (for 'finish')
  - a transition relation $\Delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Pi \cup \{\epsilon\}) \times Q$
- This defines a many-step transition relation $\hat{\Delta} \subseteq Q \times \Sigma^* \times \Pi^* \times Q$
  - $(q, x, y, q') \in \hat{\Delta}$ means "starting from state q, the input string x can be translated into the output string y, ending up in state q'." (Details omitted.)
- A finite state transducer can be run in either direction! From T you can obtain another transducer $\overline{T}$ by swapping inputs and outputs.

Stage 1: From lexical to intermediate form

- Convert 'fox+N+PL' to 'fox ^ s # '
  . . . while taking account of irregular forms like goose/geese.

- We can do this with a transducer of the following schematic form:

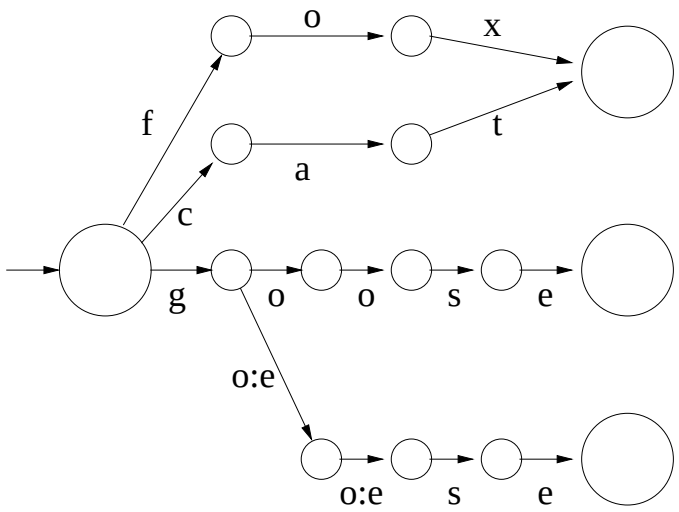


- We treat each of +N, +SG, +PL as a single symbol.

- The 'transition' labelled $+PL : \hat{}s\#$ abbreviates three transitions: $+PL : \hat{}$, $\epsilon : s$, $\epsilon : \#$.

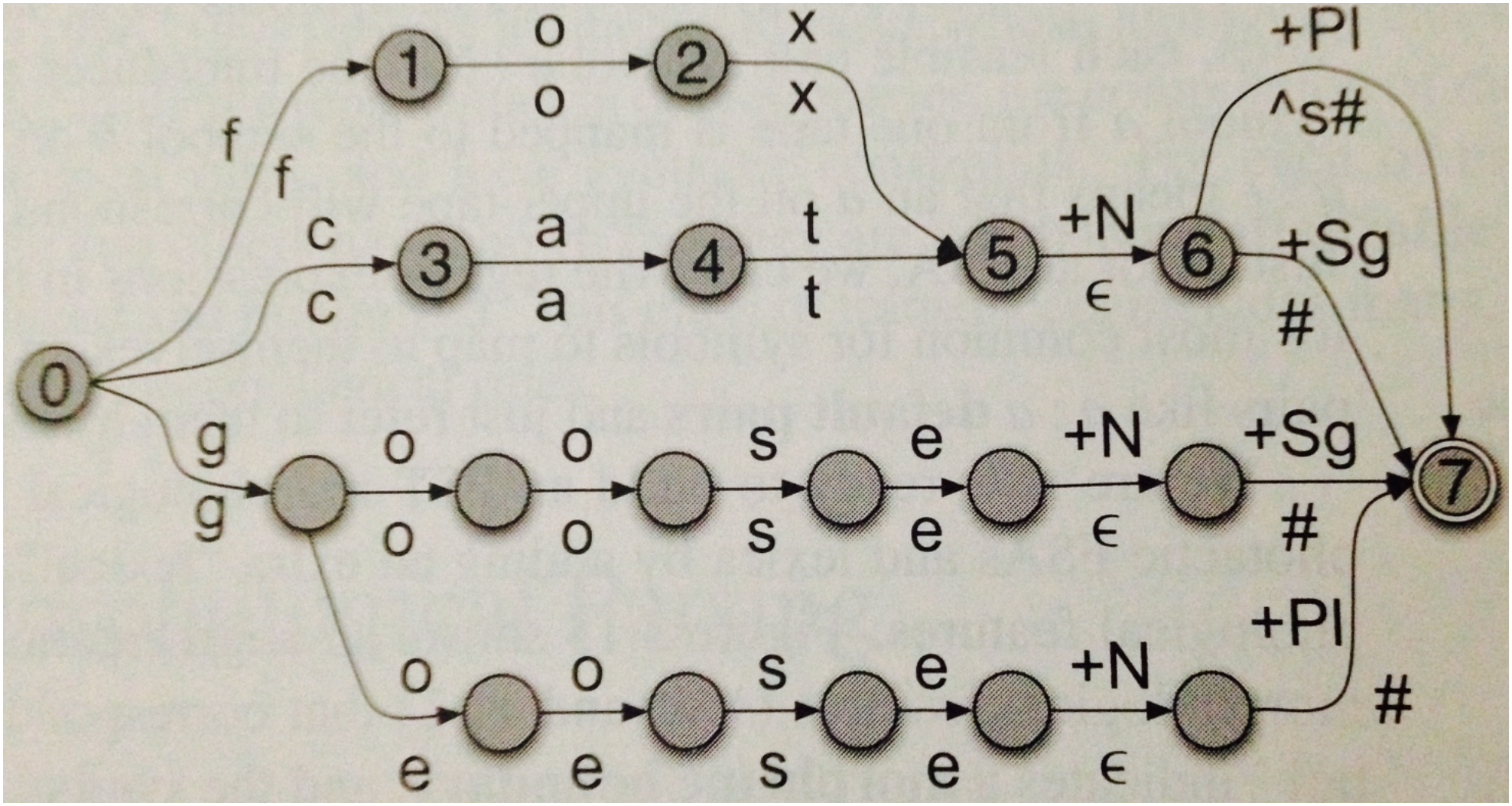
The Stage 1 transducer fleshed out

- The left hand part of the preceding diagram is an abbreviation for something like this (only a small sample shown):



- Here, for simplicity, a single label u abbreviates $u : u$.

Stage 1 in full



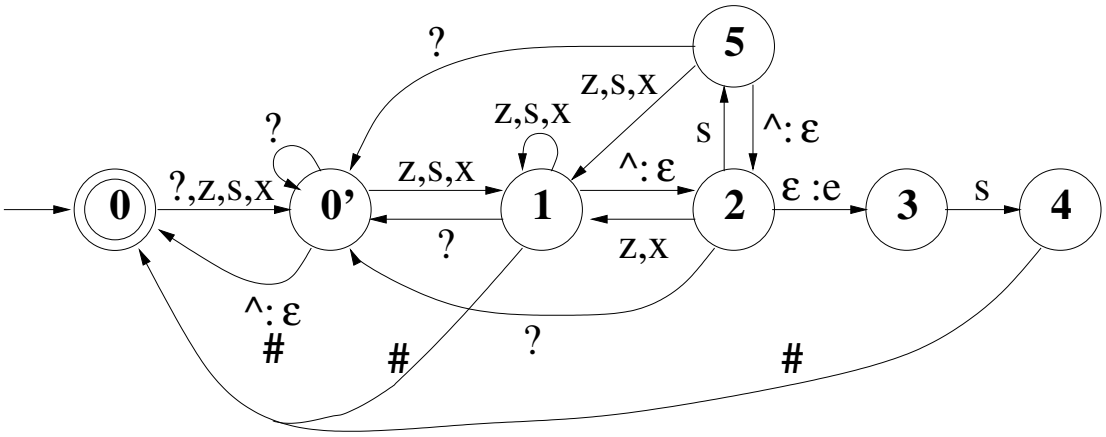
Stage 2: From intermediate to surface form

- To convert a sequence of morphemes to surface form, we apply a number of orthographic rules:
  - E-insertion: Insert e after s,z,x,ch,sh before a word-final morpheme -s.
    (fox $\rightarrow$ foxes)
  - E-deletion: Delete e before a suffix beginning with e,i.
    (love $\rightarrow$ loving)
  - Consonant doubling: Single consonants b,s,g,k,l,m,n,p,r,s,t,v are doubled before suffix -ed or -ing.
    (beg $\rightarrow$ begged)

- We shall consider a simplified form of E-insertion, ignoring ch,sh.

- This rule is oblivious to whether -s is a plural noun suffix or a 3rd person verb suffix!
  fox ^ s # $\rightarrow$ foxes

A transducer for E-insertion (adapted from J+M)



Here ? may stand for any symbol except z,s,x,ˆ,#.

(Treat # as a 'visible space character'.)

At a morpheme boundary following z,s,x, we arrive in State 2.

If the ensuing input sequence is s#, our only option is to go via states 3 and 4. Note that there's no #-transition out of State 5.

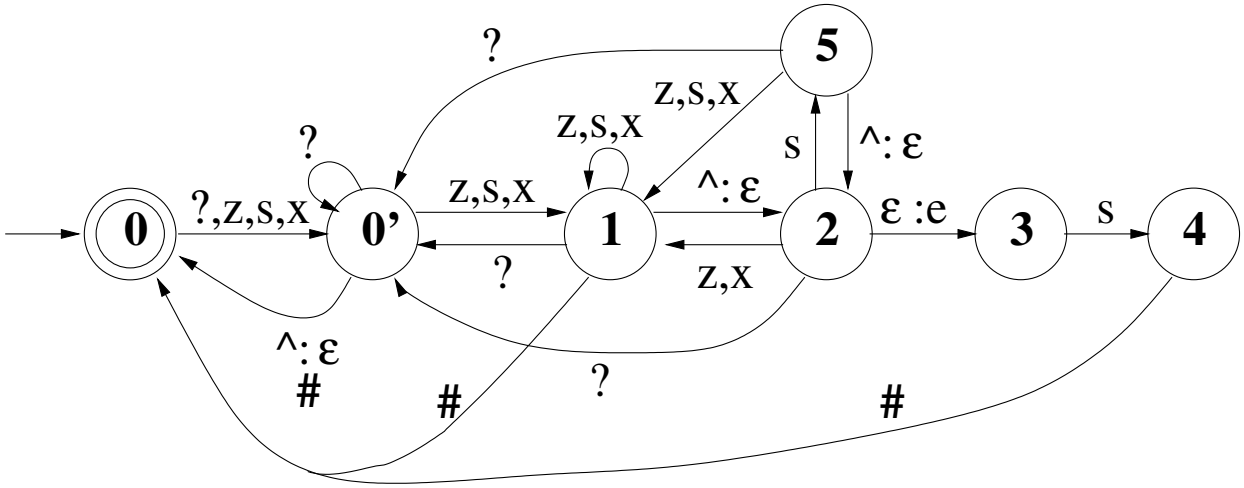
State 5 allows e.g. 'exˆserviceˆmen#' to be translated to 'exservicemen'.

Putting it all together

- FSTs can be cascaded: output from one can be input to another.
- For generation:
  - Stage 1: lexical to intermediate form; then
    Stage 2 (orthographic rules): intermediate to surface form.

  This is typically deterministic (the lexical form yields unique surface form), even though the transducers make use of non-determinism along the way.
- Running the same cascade backwards yields parsing: surface to lexical form.
- Because of ambiguity, this process is frequently non-deterministic:
  e.g. foxes might be analysed as fox+N+PL or fox+V+Pres+3SG.
- Such ambiguities are not resolved by morphological parsing itself: left to a later processing stage.

Example: 10 ways to parse asses!



On the input string 'asses', 10 transition sequences are possible!

- $0 \xrightarrow{a} 0' \xrightarrow{s} 1 \xrightarrow{s} 1 \xrightarrow{\epsilon} 2 \xrightarrow{e} 3 \xrightarrow{s} 4$, output ass^s
- $0 \xrightarrow{a} 0' \xrightarrow{s} 1 \xrightarrow{s} 1 \xrightarrow{\epsilon} 2 \xrightarrow{e} 0' \xrightarrow{s} 1$, output ass^es
- $0 \xrightarrow{a} 0' \xrightarrow{s} 1 \xrightarrow{s} 1 \xrightarrow{e} 0' \xrightarrow{s} 1$, output asses
- $0 \xrightarrow{a} 0' \xrightarrow{s} 1 \xrightarrow{\epsilon} 2 \xrightarrow{s} 5 \xrightarrow{\epsilon} 2 \xrightarrow{e} 3 \xrightarrow{s} 4$, output as^s^s
- $0 \xrightarrow{a} 0' \xrightarrow{s} 1 \xrightarrow{\epsilon} 2 \xrightarrow{s} 5 \xrightarrow{\epsilon} 2 \xrightarrow{e} 0' \xrightarrow{s} 1$, output as^s^es
- $0 \xrightarrow{a} 0' \xrightarrow{s} 1 \xrightarrow{\epsilon} 2 \xrightarrow{s} 5 \xrightarrow{e} 0' \xrightarrow{s} 1$, output as^ses
- Four of these can also be followed by $1 \xrightarrow{\epsilon} 2$ (output ^).

The Porter Stemmer

- Lexicon can be quite large with finite state transducers
- Sometimes need to extract the stem in a very efficient fashion (such as in IR)
- The Porter stemmer: a lexicon-free method for getting the stem of a given word:
  - ATIONAL $\rightarrow$ ATE (e.g., relational $\rightarrow$ relate)
    ING $\rightarrow \epsilon$ if stem comtains a vowel (e.g. motoring $\rightarrow$ motor)
    SSES $\rightarrow$ SS (e.g., grasses $\rightarrow$ grass)
- Makes errors:
  - organization $\rightarrow$ organ (computerization $\rightarrow$ computer)
    policy $\rightarrow$ police (juicy $\rightarrow$ juice)

Learning Morphological Parsers

- A vibrant area of study
- Two main paradigms: unsupervised and supervised.
- Results make errors!
  - Very good for English
    For other languages a long way to go!

  Sometimes ambiguity can't be resolved.

Summary

- Words have structure: stem + affixes
- Affixes can carry meaning and affect grammatical category.
- Languages vary on how productive, and extensive, morphology is.
- Non-deterministic Finite State Transducers (NFTs) can parse and generate morphologically complex words.
  - In effect, a compact repesentation of millions of possible word forms!
- NFTs aren't designed to resolve morphological ambiguities (no representation of context).
- Whether foxes is fox+N+PL or fox+V+3SG depends on words in its neighbourhood. . .

Next Time: Part-of-speech tagging