

# Morphology: Word Formation, FSAs and FSTs

COMP-599

Sept 10, 2015

# Compling Meetings

---

Mondays 2pm-4:30pm in this room:

- Reading groups
- Presentations
- Tutorials
- Research skills

All are welcome! Send an e-mail to [compling-owner@cs](mailto:compling-owner@cs) to get on the mailing list, or just show up.

# Textbook

---

An e-book of Jurafsky and Martin 1e is available through the McGill library.

One copy of Jurafsky and Martin 2e is available for 24h reserve loan.

I'll assign readings and practice problems from 2e, and 1e *if I can (though not guaranteed)*.

# Review

---

Match the following terms to their object of study

Semantics

Phonetics

Discourse

Syntax

Phonology

Pragmatics

Morphology

Sound patterns

Literal meaning

Word structure

Implied meaning

Speech sounds

Sentence structure

Passage structure

# Outline

---

English morphology

    Inflectional and derivational morphology

Lemmatization vs. stemming

Formalization as FSA, FST

# Starting Small

---

We begin by starting from the smallest level of grammatical unit in language, the **morpheme**.

*anti- dis- establish -ment -arian -ism*

Six morphemes in one word

*cat -s*

Two morphemes in one word

*of*

One morpheme in one word

# Types of Morphemes

---

## Free morphemes

May occur on their own as words (*happy, the, robot*)

## Bound morphemes

Must occur with other morphemes as parts of words

Most bound morphemes are **affixes**, which attach to other morphemes to form new words.

**Prefixes** come before the **stem**: *un-*, as in unhappy

**Suffixes** come after the stem: *-s*, as in robots

**Infixes** go inside: *-f\*\*king-*, as in absof\*\*king-lutely

(Not really an infix, but as close as we get in English)

**Circumfixes** go around: *em- -en*, as in embolden

# Derivation vs. Inflection

---

**Inflectional** morphology is used to express some kind of grammatical function required by the language

*go -> goes*                      *think -> thought*

**Derivational** morphology is used to derive a new word, possibly of a different part of speech

*happy -> happily*    *establish -> establishment*

**Exercise:** come up with three prefixes and suffixes in English. Make sure to include at least one derivational and one inflectional affix.



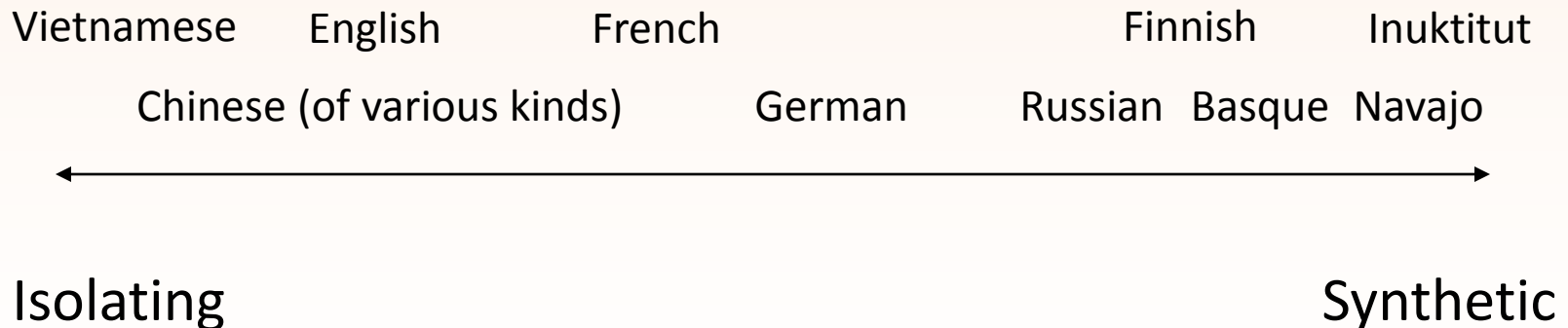
# Cross-linguistic Variation

---

Languages across the world vary by their morpheme-to-word ratio.

**Isolating** languages: low morpheme-to-word ratio

**Synthetic** languages: high morpheme-to-word ratio



Note: this chart is only a rough guide, not a precise ranking!



# In Other Words...

---

English morphology is relatively boring!

(And this is why we're only spending one class on it.)

# But It Still Matters!

---

Recognize whether a word is actually English

*foxes* vs. *\*foxs*

Abstract away details that don't matter for an application

*The campers saw a bear.*

*The campers see a bear.*

*The camper saw a bear.*

- In all cases, a bear was seen!

Generate the correct form of a word

*see +PRESENT +3SG -> sees*

*see +PAST +2PL -> saw*

# Computational Tasks

---

## Morphological recognition

Is this a well formed word?

## Stemming

Cut affixes off to find the stem

- *airliner* -> *airlin*

## Morphological analysis

**Lemmatization** – remove inflectional morphology and recover lemma (the form you'd look up in a dictionary)

- *foxes* -> *fox*

Full morphological analysis – recover full structure

- *foxes* -> *fox* +N +3SG

# Morphological Recognition

---

Are these valid English words?

*friendship, unship, defriender, friendes*

Relevant issues:

What prefixes and suffixes can go with a word?

- *-ship, un-, de-, -s*

Different forms of an affix

- *fox -> foxes*
- *friend -> friends*
- *fly -> flie*s

Exceptions

- *goose -> geese*

# Lexicon

A list of all words, affixes and their behaviours. Entries are often called **lexical items** (a.k.a., lexical entries, lexical units).

- e.g., Noun declensions (Jurafksy and Martin, p. 54)

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

# Morphotactics

Tells us the sequence in which morphemes may be combined.

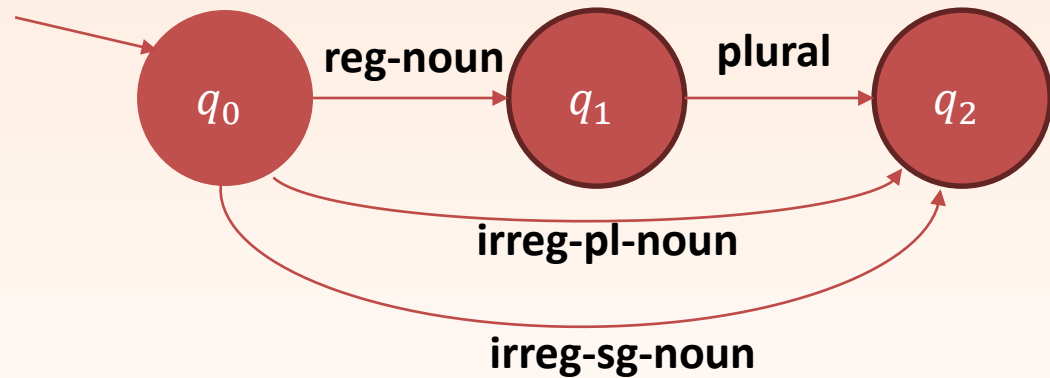
- e.g., English nouns:

reg-noun

reg-noun + -s

irreg-sg-noun

irreg-pl-noun



Nodes with outline represent an **accepting state**. (This is a word)

This is a representation of a **finite state automaton (FSA)**.



# Finite State Automata

---

A model of computation that takes in some input string, processes them one symbol at a time, and either **accepts** or **rejects** the string.

- e.g., we write a FSA to accept only valid English words.

A particular FSA defines a **language** (a set of strings that it would accept).

- e.g., the language in the FSA we are writing is the set of strings that are valid English words.

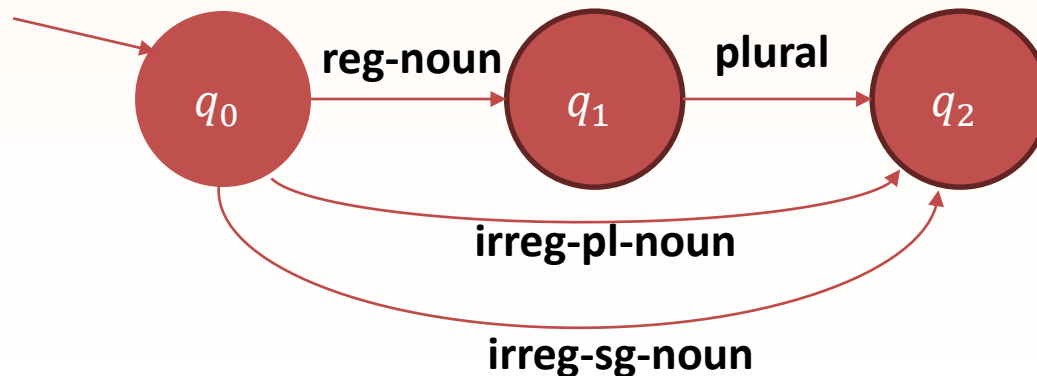
The set of languages for which there is an FSA that could describe that language is called **regular languages**.

# Definition of FSA

A FSA consists of:

- $Q$  finite set of states
- $\Sigma$  set of input symbols
- $q_0 \in Q$  starting state
- $\delta : Q \times \Sigma \rightarrow Q$  transition function from current state and input symbol to next state
- $F \subseteq Q$  set of accepting, final states

Identify the above components:



# Expanding the FSA

---

Use lexicon to expand the morphotactic FSA into a character-level FSA

# Exercise

---

Extend the previous FSA to account for *regular* orthographic variations of the plural suffix.

- Ends with consonant + *y*: replace *y* with *ies*
  - e.g., *pony*, *sky* but not *boy*
- Ends with *-s*, *-z*, *-x*, *-ch*, *-sh* -> add *-es*
  - e.g., *kiss*, *dish*, *witch*

Check your FSA by seeing whether it correctly accepts English words that you model and rejects those that are not

# Stemming - Porter Stemmer



An ordered list of rewrite rules to approximately recover the stem of a word (Porter, 1980)

- Basic idea: chop stuff off and glue some endings back on
- Not perfect, but sometimes results in a slight improvement in downstream tasks
- Advantage: no need for lexicon

# Examples of Porter Stemmer Rules

---

*ies -> i*

- *ponies -> poni*

*ational -> ate*

- *relational -> relate*

If word is long enough (# of syllables, roughly speaking),

*al -> ε*

- *revival -> reviv*

# Morphological Parsing

---

Recover an analysis of the word structure

- *foxes* -> *fox +N +PL*
- *foxes* -> *fox +V +3SG.PR*

In fact, we will add an intermediate layer for convenience:

Surface	<i>foxes</i>	<i>cat</i>
Intermediate	<i>fox^s#</i>	<i>cat^s#</i>
Underlying	<i>fox +N +PL</i>	<i>cat +N + PL</i>

- Lets us not have to deal with intricacies in the orthographic rules at the same time as the rest
- Irregular nouns handled by intermediate->underlying step

# Expanded Lexicon

Basic idea: add more annotations to the lexicon

Map surface to intermediate level:

reg-noun	irreg-pl-noun	irreg-sg-noun	plural
<i>fox</i>	<i>g e:o e:o se</i>	<i>goose</i>	<i>s:^s</i>
<i>cat</i>	<i>sheep</i>	<i>sheep</i>	
<i>aardvark</i>	<i>m i:o ε:u c:s e</i>	<i>mouse</i>	

## Notation:

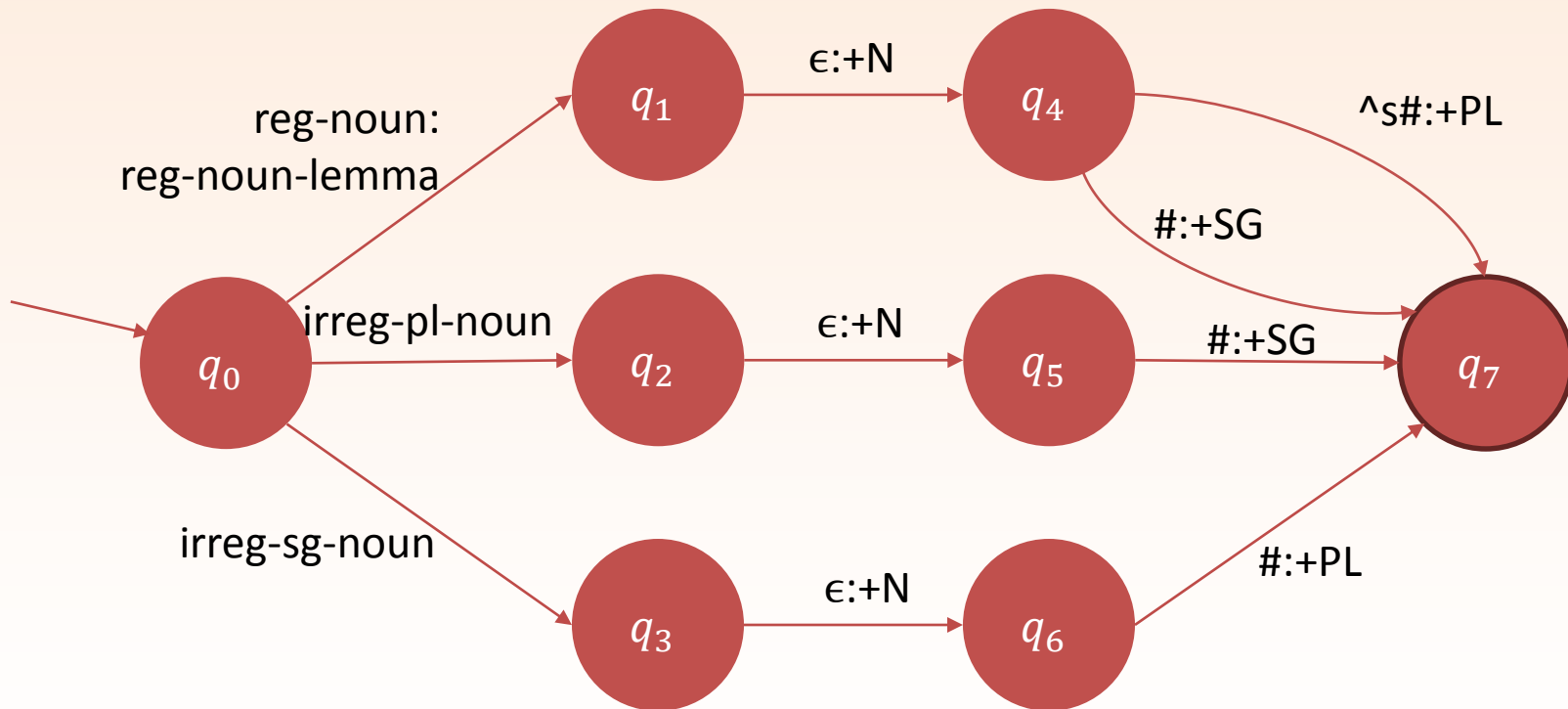
- Single letter: map a letter to itself
- Implicit  $\epsilon:\#$  transition at ends of words
- Note: in J&M p. 62, they wrote the table for generation, so the letters are flipped



# Finite State Transducers

Next step: Intermediate to underlying level

- Need to expand parts like “reg-noun” below with lexicon



Question: what happens with *sheep*?

# Exercise

---

Change our previous character-level FSA:

- add outputs
- add states for the necessary adjustments as necessary
- i.e., this should map from the surface to the intermediate level

Now that we have added outputs, the machine is no longer a FSA. It is a **Finite State Transducer**.

# Definition of FST

---

A FST consists of:

- $Q$  finite set of states
- $\Sigma, \Delta$  sets of input symbols, output symbols
- $q_0 \in Q$  starting state
- $\delta : Q \times \Sigma \rightarrow Q$  transition function from current state and input symbol to next state
- $\sigma : Q \times \Sigma \rightarrow \Delta$  output function from current state and input symbol to output symbol
- $F \subseteq Q$  set of accepting, final states

Identify the above components in the previous FST

# Composing FSTs

---

We have two FSTs:

1. Surface to intermediate      FST1    *fox -> fox#*
2. Intermediate to underlying    FST2    *fox# -> fox +N +SG*

**Compose** them to make full morphological parser

- surface -> FST1 -> intermediate -> FST2 -> underlying

The composed machine is also a FST.

# Inverting FSTs

---

We now have a FST for morphological parsing. What about morphological generation?

- Simply flip input and output symbol!  
     $\wedge s\# : +PL$  becomes  $+PL : \wedge s\#$
- underlying  $\rightarrow$  FST<sup>-1</sup><sub>2</sub>  $\rightarrow$  intermediate  $\rightarrow$  FST<sup>-1</sup><sub>1</sub>  $\rightarrow$  surface

# Overall Picture

---

Build a lexicon of the words you care about

- Handle regular orthographic variations using this intermediate representation – write some rules or FSTs to describe them
- Handle irregular words by building exception lists

The multiple FSTs that you write will be combined in various ways to produce the final morphological analyzer/generator:

- Composition, intersection, ...

In general, FSAs and FSTs are very useful models that pop up in many areas of NLP!

# Exercise

---

Write a (surface to intermediate) transducer to segment monosyllabic verbs. Deal with the *-ed* and *-ing* suffixes.

- e.g., *jump* -> *jump#*      *jumped* -> *jump^ed#*  
*jumping* -> *jump^ing#*
- Include some irregular verbs (e.g., *see*, *hit*)
- Deal with verbs that end in *e* (e.g., *hope*, *hate*)
- Deal with consonant doubling – when in a CVC pattern
  - e.g., *chat* -> *chatted*, *chatting*    *bat* -> *batted*, *batting*  
[just handle the case of *t*]

Then, write the template of the intermediate to underlying transducer as in slide 25.