Ling/CSE 472: Introduction to Computational Linguistics

4/6
Morphology and FST

Overview

- Morphology primer
- Using FSAs to recognize morphologically complex words
- FSTs (definition, cascading, composition)
- FSTs for morphological parsing
- Reading questions

Morphology primer

- Words consist of stems and affixes.
- Affixes may be prefixes, suffixes, circumfixes or infixes.
    - Examples?
- (Also: root and pattern morphology)
    - Examples?
- Phonological processes can sometimes apply to combinations of morphemes.

Phonology at morpheme boundaries

Examples:

English	/s/	Spanish	/s/
Singular	Plural	Singular	Plural
[kæt] ‘cat’	[kæts] ‘cats’	[ninjo] Eng: ‘boy’	[ninjos]
[dɑg] ‘dog’	[dɑgz] ‘dogs’		
[finʧ] ‘finch’	[finʧəz] ‘finches’	[kaɾakol] Eng: ‘snail’	[kaɾakoles]

More on morphology

- Languages vary in the richness of their morphological systems.
- Languages also vary in the extent to which phonological processes apply at (and sometimes blur) morpheme boundaries.
- English has relatively little inflectional morphology, but fairly rich (if not perfectly productive) derivational morphology.
- Turkish has far, far more. Sak et al (2011) estimate 4.1M distinct word types in a 4.5M word corpus and 52,000 distinct lexical endings (word forms minus the stem).

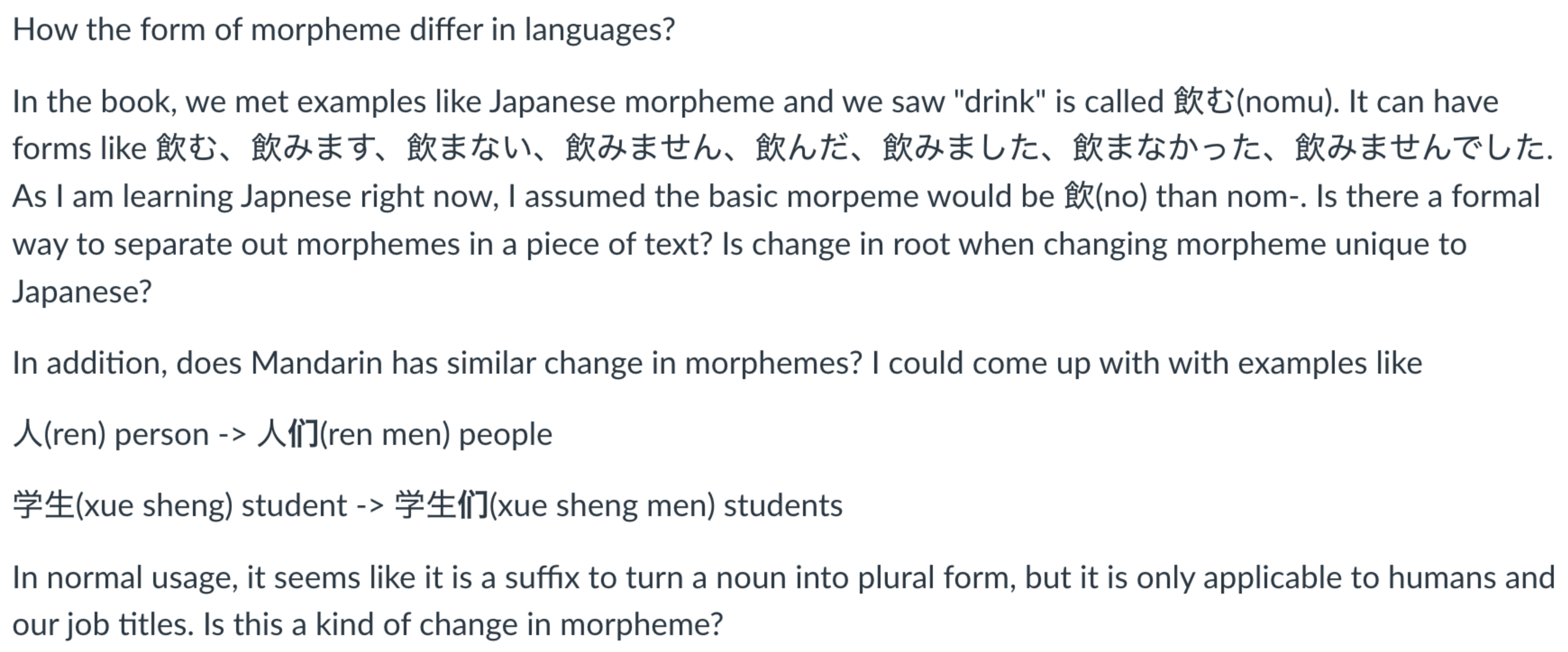
Words with lots of morphemes (use + as morpheme boundary)

Total Results: 0

Questions

- More examples of complexity in morphology?
- What underlying representations might we want?
- Why would we want to get to those underlying representations?
- How do things change when we consider orthographic rules rather than phonological rules?

Morphology reading questions



How the form of morpheme differ in languages?

In the book, we met examples like Japanese morpheme and we saw "drink" is called 飲む(nomu). It can have forms like 飲む、飲みます、飲まない、飲みません、飲んだ、飲みました、飲まなかった、飲みませんでした. As I am learning Japnese right now, I assumed the basic morpeme would be 飲(no) than nom-. Is there a formal way to separate out morphemes in a piece of text? Is change in root when changing morpheme unique to Japanese?

In addition, does Mandarin has similar change in morphemes? I could come up with with examples like

人(ren) person -> 人们(ren men) people

学生(xue sheng) student -> 学生们(xue sheng men) students

In normal usage, it seems like it is a suffix to turn a noun into plural form, but it is only applicable to humans and our job titles. Is this a kind of change in morpheme?

Morphology reading questions

- Are the wholly different inflections of verbs typically the result of a borrowing from another language or some other different kind of historical change influenced by another language? Or can those arise in similar ways as do phonological and morphological changes that are typically motivated by perceptive and articulative reasons. If not, is that why they are so rare in languages?

Morphology reading questions

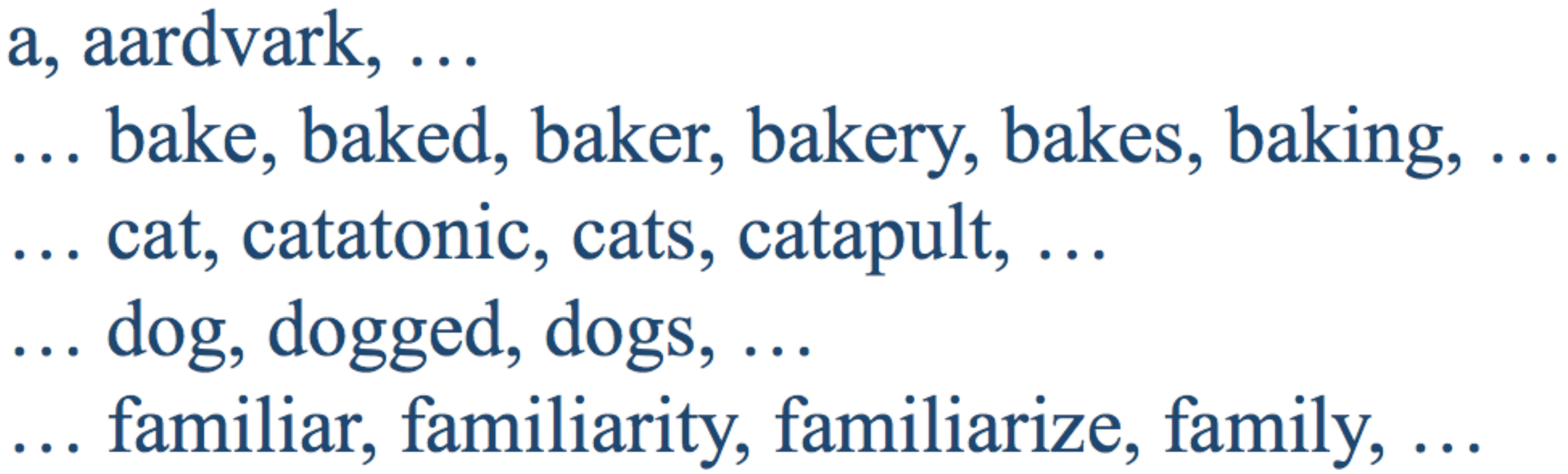
- I am wondering what ways languages handle something such as plurals without a simple letter (or variation of letters) like we do in English with the /s/, /z/, or /az/. For example, I know that indonesian duplicates nouns to represent plurals: "anak" is child while "anak-anak" is children.

Morphological parsing

- Parsing: Producing a linguistic structure for an input
- Examples of morphological parsing:
    - Separating words into stems/roots and affixes:
        - e.g. input: cats   parse output: cat +s
    - Labeling morphemes with category labels:
        - e.g. input: cats   parse output: cat +N +PL
        -   input: ate   parse output: eat +V +PAST

Can we just model a lexicon as a list?

- What about using a large list as a lexicon?



a, aardvark, …
… bake, baked, baker, bakery, bakes, baking, …
… cat, catatonic, cats, catapult, …
… dog, dogged, dogs, …
… familiar, familiarity, familiarize, family, …

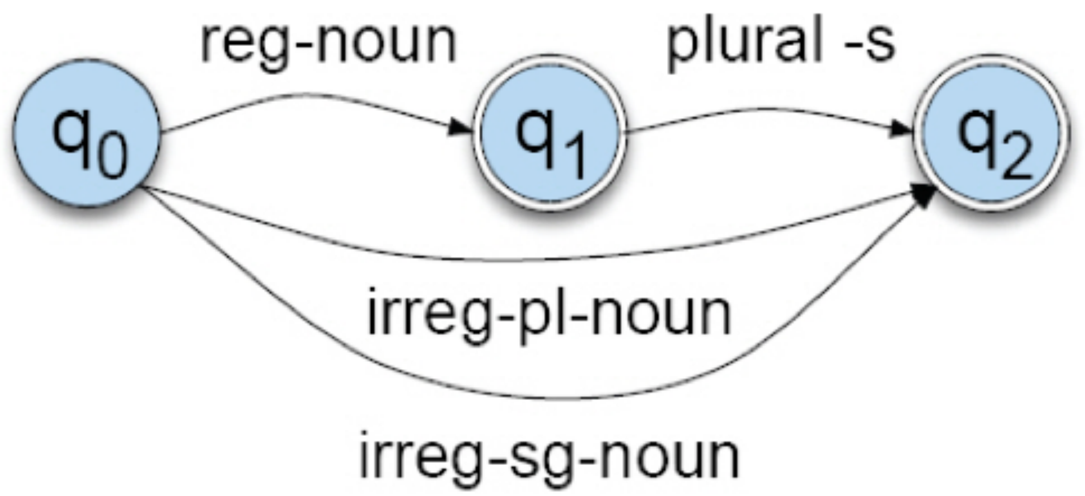
- Problem?

Using FSAs to recognize morphologically complex words

- Create FSAs for classes of word stems (word lists).
- Create FSA for affixes using word classes as stand-ins for the stem word lists.
- Concatenate FSAs for stems with FSAs for affixes.

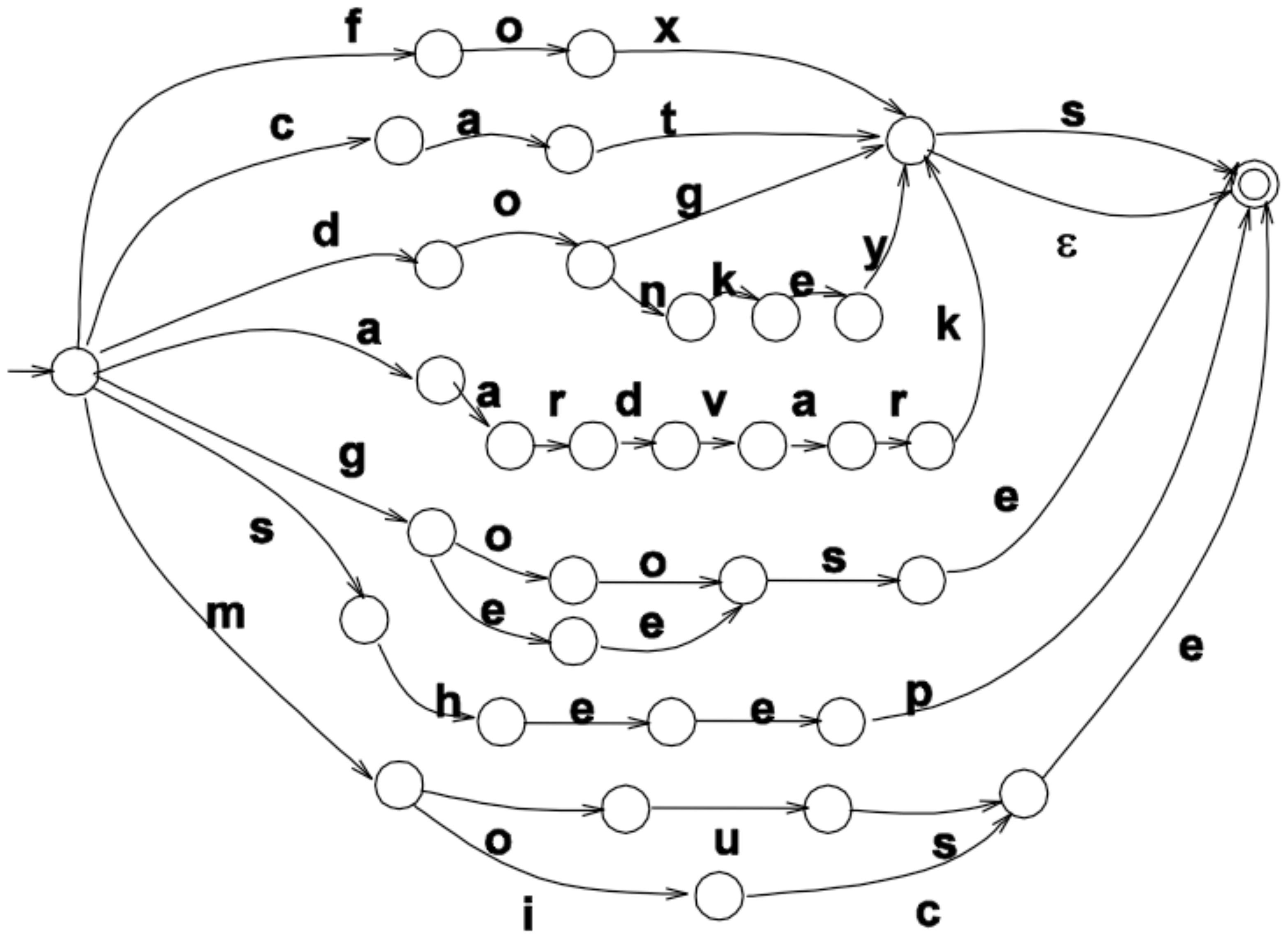
FSA Example using Word Classes

- Defining morpheme selection and ordering for singular and plural English nouns:



J&M text, Fig 3.3

A variation with some words

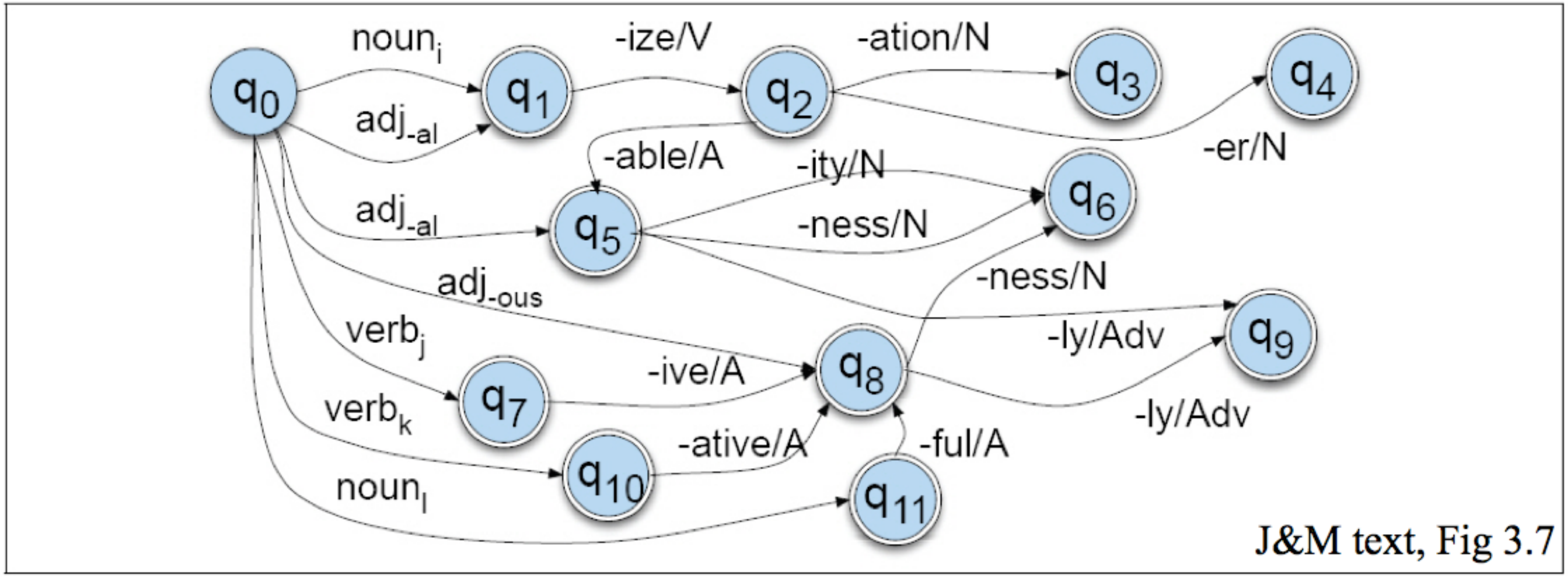


Note: Orthographic issues are not addressed.

More Generalizations

- ... formal, formalize, formalization, ...
- ... fossil, fossilize, fossilization, ...
- These represent sets of related words.
- New forms are built with the addition of derivational morphology.
    - ADJ + -ity NOUN
    - ADJ or NOUN + -ize VERB

Derivational Rules



- What strings would this recognize? Is that really what we want?

Morphological Parsing

- A parsing task:
  - – Recognize a string
  - – Output information about the stem and affixes of the string
- Something like this:
  - – Input: cats
  - – Output: cat+N+PL
- We will use Finite-State Transducers to accomplish this.

Morphological parsing reading questions

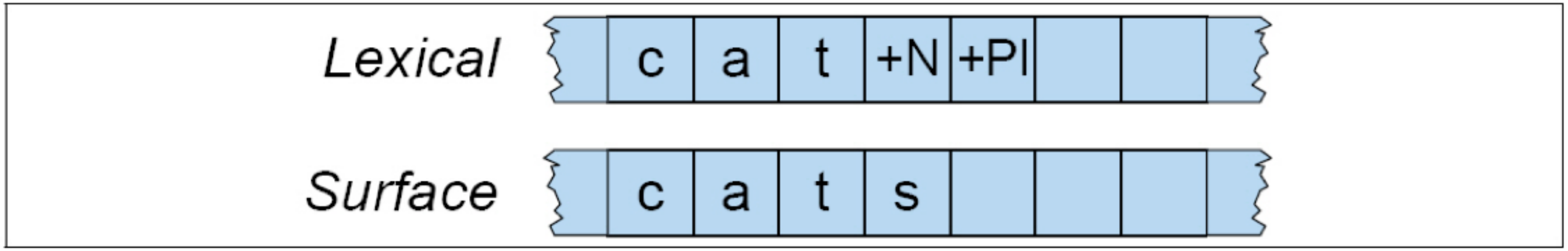
- In languages like French where verbs have lots different conjugated forms (as in Bender #25), is it easier for a NLP system to have those conjugations hard-coded? i.e. have all the forms for different person/number/tense listed out with their corresponding uses? Or could you more easily code a program which knows when to apply certain affixes? Like taking the infinitive *descendre* and turning it into *descend+ons*.

Finite-State Transducer (FST)

- An FST:
  - is like an FSA but defines regular relations, not regular languages
  - has two alphabet sets
  - has a transition function relating input to states
  - has an output function relating state and input to output
  - can be used to recognize, generate, translate or relate sets

Visualizing FSTs

- FSTs can be thought of as having an upper tape and a lower tape (output).

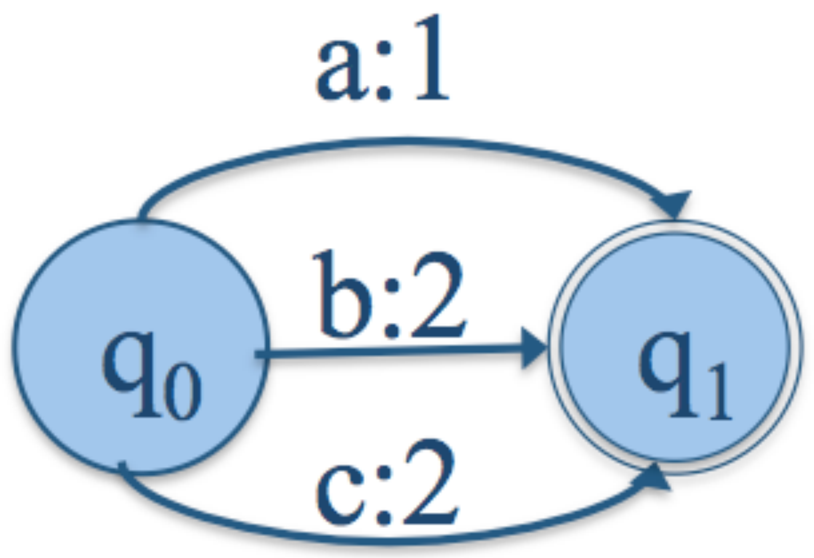


J&M text, Fig 3.12

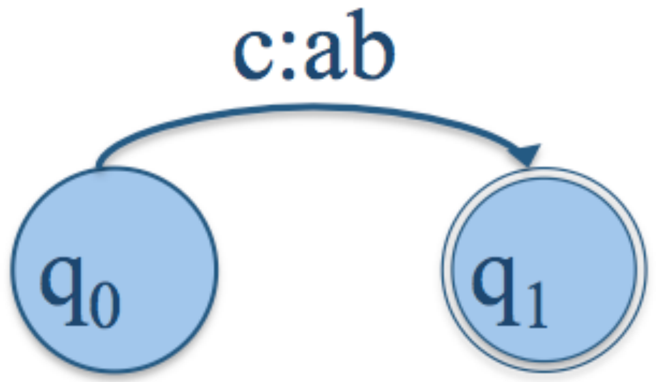
Regular Relations

- Regular language: a set of strings
- Regular relation: a set of pairs of strings
    - E.g., Regular relation = {a:1, b:2, c:2}
    - Input Σ = {a,b,c} Output ={1, 2}

FST:

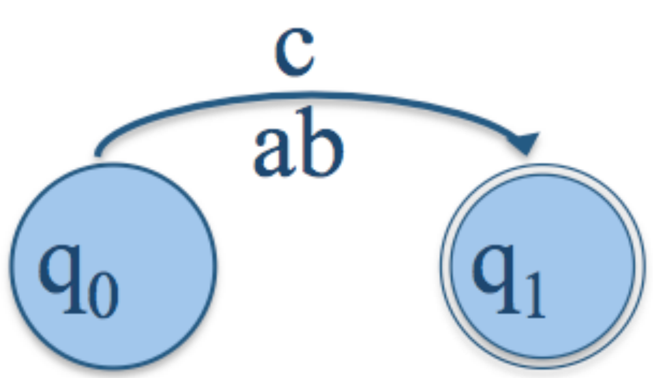


FST conventions

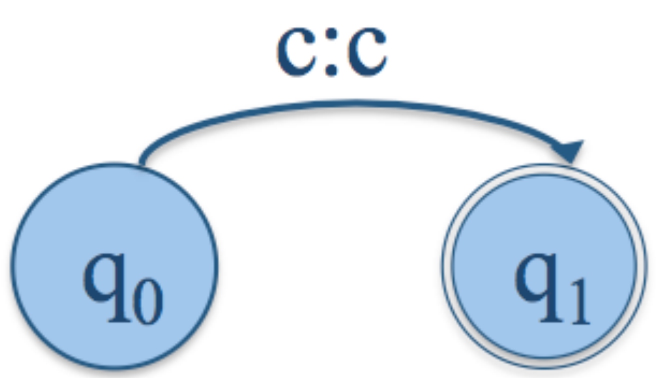


Complex input element

=

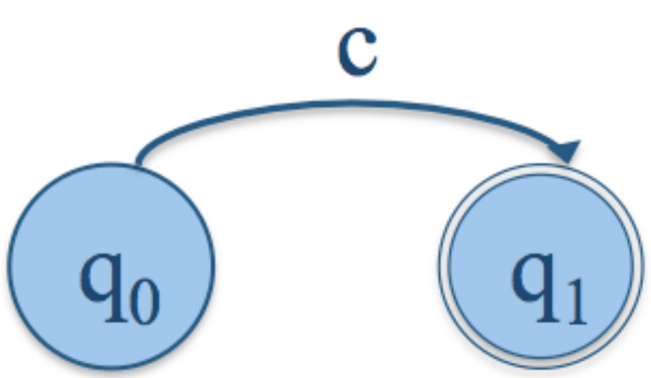


Divided into upper and lower

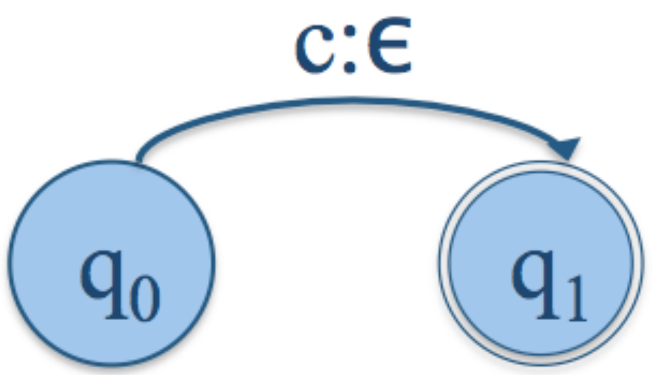


Default pair

=



Default pair - shortcut



c on upper, nothing on lower

FSTs: Not just fancy FSAs

- Regular languages are closed under difference, complementation and intersection; regular relations are (generally) not.
- Regular languages and regular relations are both closed under union.
- But regular relations are closed under composition and inversion; not defined for regular languages.

Inversion

- FSTs are closed under inversion, i.e., the inverse of an FST is an FST.
- Inversion just switches the input and output labels.
  - e.g., if T1 maps ‘a’ to ‘1’, then T1-1 maps ‘1’ to ‘a’
- Consequently, an FST designed as a parser can easily be changed into a generator.

Composition

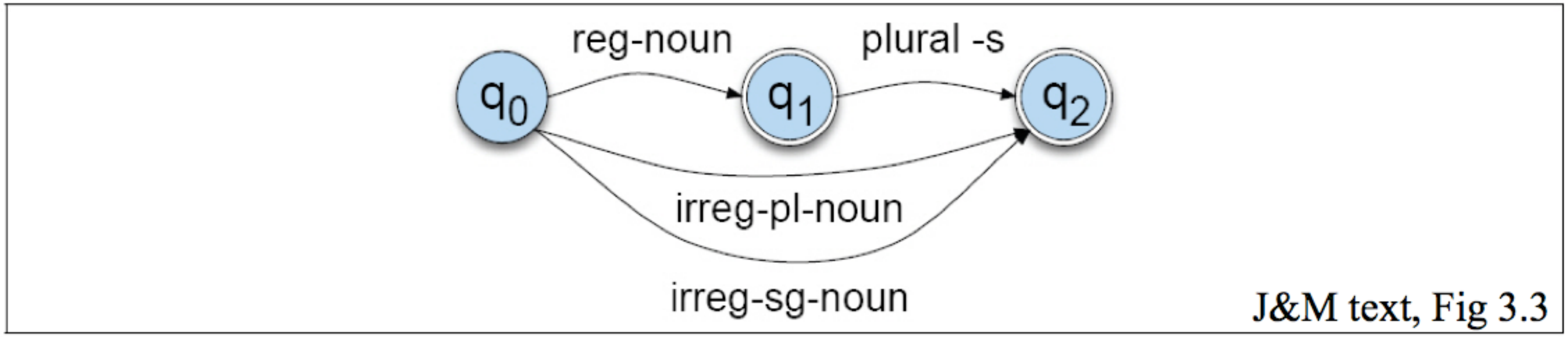
- It is possible to run input through multiple FSTs by using the output of one FST as the input of the next. This is called Cascading.
- Composing is equivalent in effect to Cascading but combines two FSTs and creates a new, more complex FST.
- $T1 \circ T2 = T2\,(T1(s))$
  - where s is the input string

Composition Example

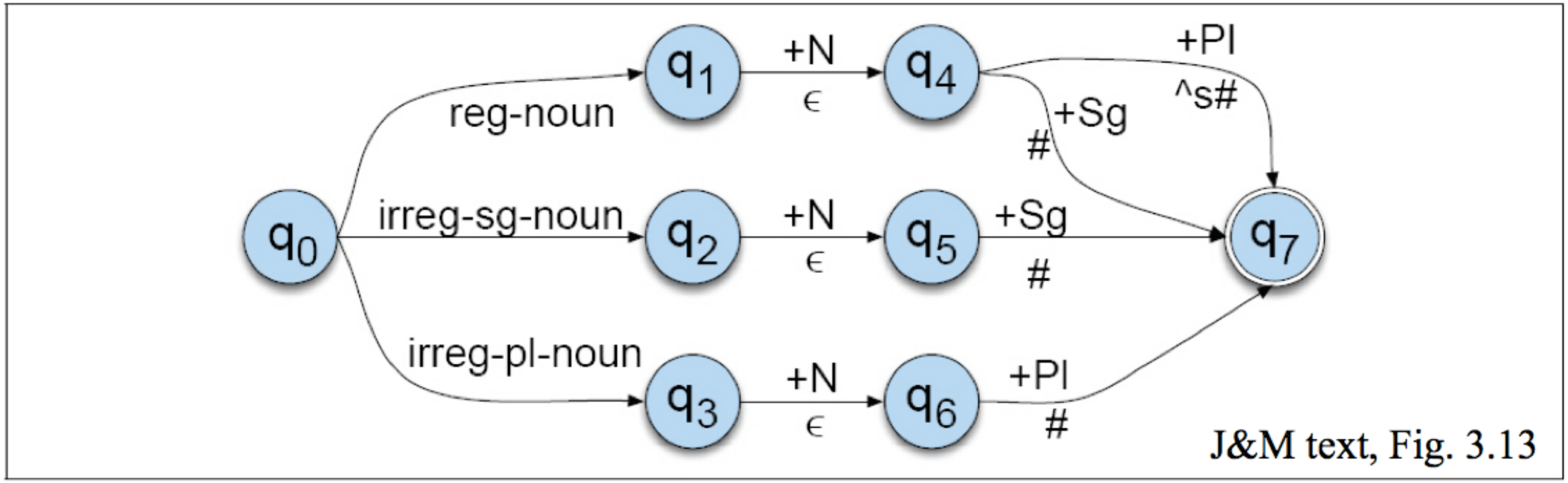
- Very simple example:
  - T1 = {a:1}
  - T2 = {1:one}
  - T1 ◦ T2 = {a:one}
  - T2(T1(a)) =one
- Note that order matters: T1(T2(a)) ≠ one
- Composing will be useful for adding orthographic rules.

Comparing FSA Example with FST

- Recall this FSA singular and plural recognizer:



An FST to parse English Noun Number Inflection



^ = morpheme boundary
= word boundary

What are the benefits of this FST over the previous FSA?
What is the input alphabet? What does the output look like?

Review: FSAs and FSTs

- FSAs define sets of strings (regular languages).
- FSTs define sets of ordered pairs of strings (regular relations).
- Formally interesting because not all languages/relations can be defined by FSAs/FSTs.
- Are all finite languages and relations regular?
- Linguistically interesting because:
  - FSAs have enough power for morphotactics.
  - FSTs have (almost) enough power for morphophonology.
  - Both are very efficient.

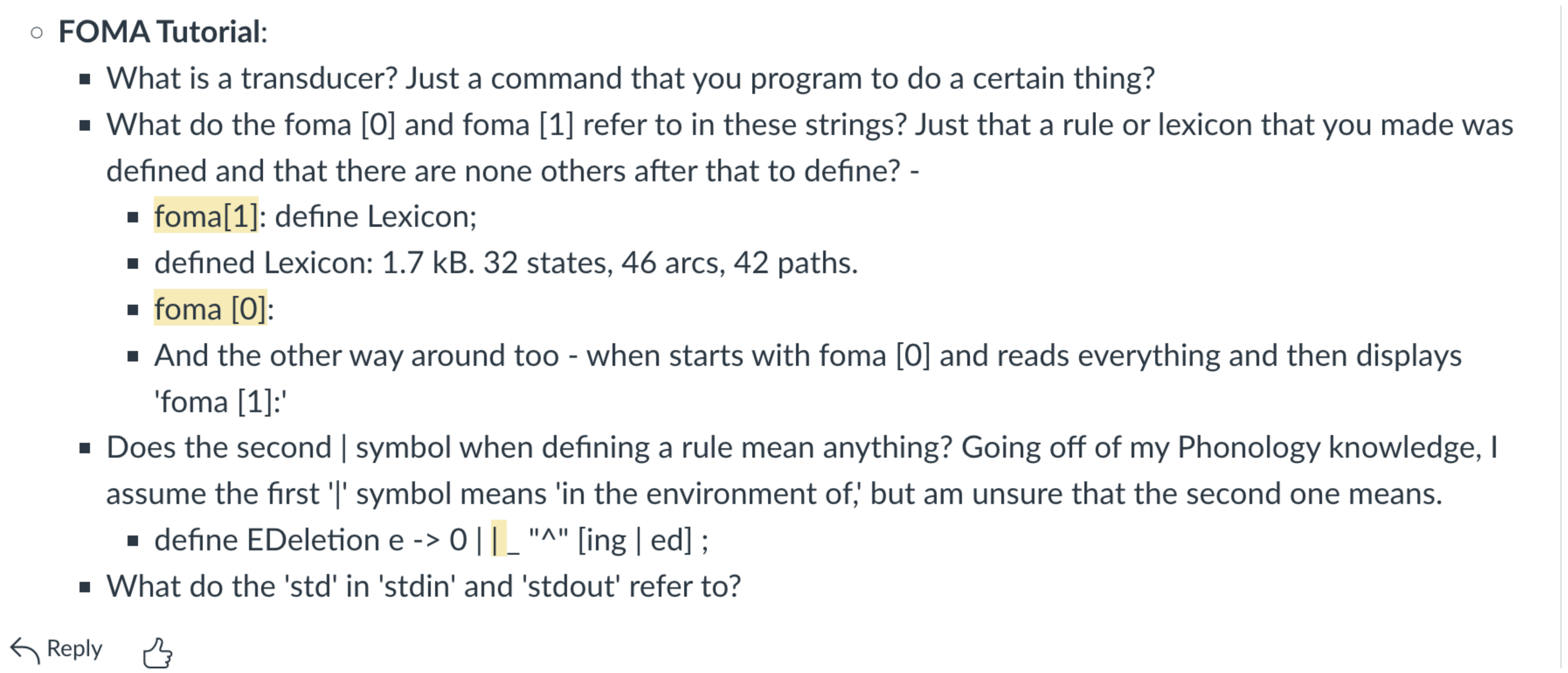
FSTs: “Quiz”

- Why do FSTs have complex symbols labeling the arcs?
- What happens if you give an FST an input on only one “tape”?
- What happens if the input has symbols outside the FST’s alphabet?
- Do the upper and lower tape strings always have the same length?

foma/xfst regex syntax

- Why is it so different from what we see in J&M and elsewhere?
- Why are there so many operators?

foma reading questions



- **FOMA Tutorial**:
  - What is a transducer? Just a command that you program to do a certain thing?
  - What do the foma [0] and foma [1] refer to in these strings? Just that a rule or lexicon that you made was defined and that there are none others after that to define? -
    - foma[1]: define Lexicon;
    - defined Lexicon: 1.7 kB. 32 states, 46 arcs, 42 paths.
    - foma [0]:
    - And the other way around too - when starts with foma [0] and reads everything and then displays 'foma [1]:'
  - Does the second | symbol when defining a rule mean anything? Going off of my Phonology knowledge, I assume the first '|' symbol means 'in the environment of,' but am unsure that the second one means.
    - define EDeletion e -> 0 | | _ "^" [ing | ed] ;
  - What do the 'std' in 'stdin' and 'stdout' refer to?

Reply

Overview

- Morphology primer
- Using FSAs to recognize morphologically complex words
- FSTs (definition, cascading, composition)
- FSTs for morphological parsing
- Reading questions

Reading questions

- What uses do morphological analyzers have?

Reading questions

- Could word forms with ambiguous morphology like "watches" and "runs" could cause problems when using a morphological analyzer or if this is never really an issue.

- How are words with multiple outputs (like run+V+3p+Sg and run+N+Pl for "runs") handled by other programs/functions?

- I don't really get how using FSTs for morphological analysis could accurately deal with ambiguities like humans do. My assumption is that analyzers don't need to "deal" with ambiguities in their subtask, but instead just outputs all possible interpretations and let something else make the choice to determine which to use in a sentence.

Reading questions

- Would love to learn more about the use of finite-state transducers as generative models when inverted. As the morphological rules are encoded within the transitions, is the idea to start with a set of individual morphemes and end up with a complete word as the output of the model?

Reading questions

- How flexible are FSTs with regards to feeding the model with novel words that still follow the morphological rules (e.g. words that don't really exist but sound real: "Twas brillig, and the slithy toves, Did gyre and gimble in the wabe"). When introduced to the model, are we still able to determine the constituent morphemes?

- Can we put some random morphemes together through an inverted FST to create novel words with new meanings?

Reading questions

- How can finite state transducers be used to analyze and generate morphological patterns in non-standard varieties of a language, such as dialects or sociolects? For example, in Appalachian English someone might say seed instead of saw, using -ed as a past tense marker.

- What exactly is a transducer in the context of the FOMA Tutorial? What is its exact function?

Reading questions

- Instead of listing each sibilant separately in the e-insertion rule, for example, is there a way to input the distinctive features of these phonemes and have the rule apply to characters that correspond with these distinctive features? Or does that pose issues for when the orthography does not correspond with the phonetic realization of those sounds?

Reading questions

- Are irregularities in V and N inflections rare enough that it is not inefficient to list them out separately?

- I'm confused about the example of a priority union in the reading. Is there a connection between "cat+N+P1 --> cats" and "make+V+Past --> makes"?

- I am very confused by the lexicon script. I don't really know what is happening when the transition to a lexicon Ninf is made. Also what does it mean to put the lexicon on the stack?

NLP/compling in the news

- https://www.smbc-comics.com/comic/ethics-4