CMPT 413
Computational Linguistics

Anoop Sarkar

http://www.cs.sfu.ca/~anoop

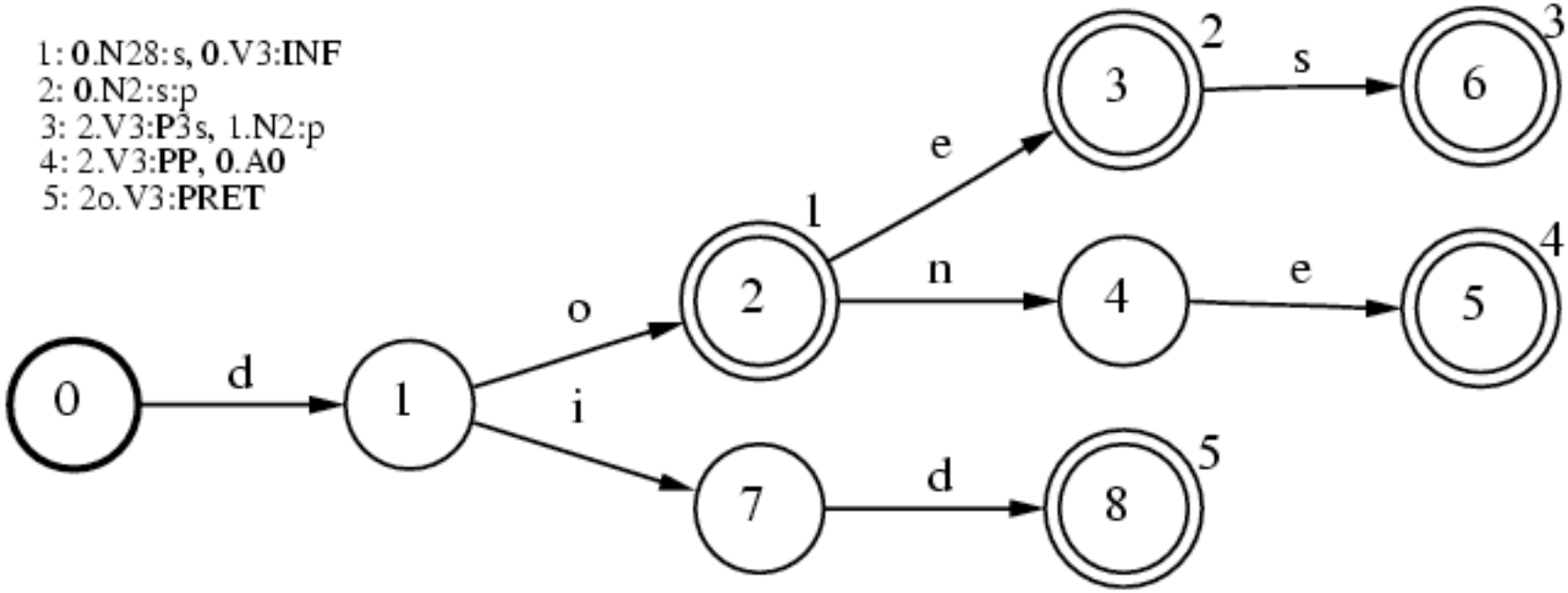
Algorithms for FSMs
(finite-state machines)

- Recognition of a string in a regular language: is a string accepted by an NFA?
- Conversion of regular expressions to NFAs
- Determinization: converting NFA to DFA
- Converting an NFA into a regular expression
- Other useful *closure* properties: union, concatenation, Kleene closure, intersection

The Lexicon

- Assume we process natural language input in the form of sequences of words
- The sequence is processed to compute some meaningful response or translation
- The role of the **lexicon** is to associate linguistic information with words of the language
- Many words are ambiguous: with more than one entry in the lexicon
- Information associated with a word in a lexicon is called a **lexical entry**

Lexicons stored as FSMs



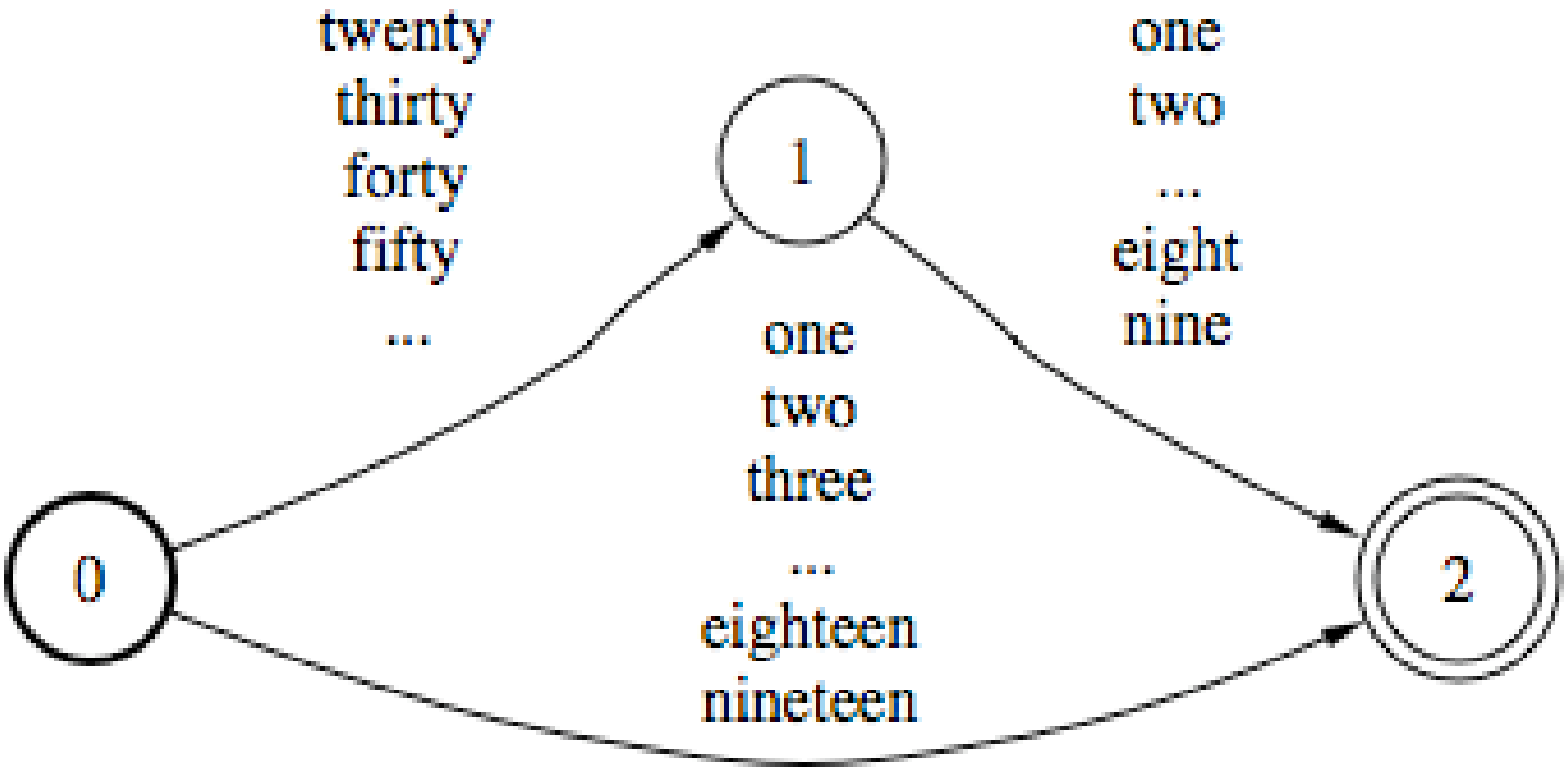
done, 4:- do:V+PP (*the turkey was done*); done:A (*a done deal*)
doe, 3:- doe: N (*a deer, a female deer; U.S. Dept. of Energy*)
...

Lexicons stored as FSMs

DICTIONARIES	Name	FDELAF French V.7	FDELACF Compound	GDELAF General	EDELAF English	IDELAF Italian
	Nb of lines	672,000	156,000	828,000	145,000	612,000
	Initial size	21.2 Mb	5.6 Mb	27.9 Mb	3.6 Mb	20 Mb
AUTOMATA	Nb of states	84,600	322,800	389,650	48,770	78,320
	Nb of transitions	181,910	466,570	633,520	101,970	177,350
	Size	2.4 Mb	7.6 Mb	10 Mb	1.2 Mb	2.3 Mb
	Compacted size	818 Kb	1.88 Mb	2.70 Mb	447 Kb	806 Kb
	Nb of codes	13,200	1,750	14,950	1,590	11,190
	Size of codes	358 Kb	445 Kb	403 Kb	25 Kb	257 Kb
	Total final size	1.2 Mb	1.9 Mb	3.1 Mb	470 Kb	1.1 Mb
TIME SPENT	Constr. (CRAY)	-	12h40	18h53	-	-
	Constr. (HP)	12'30	-	-	4'55"	12'30
	Constr. (NEXT)	1h18'	-	-	17'	1h20'
	Look-up (HP)	90 w/ms	90 w/ms	90 w/ms	90 w/ms	90 w/ms

from (Mohri, 1995)

Simple NL grammars in FSMs



Numbers 1-99 in English

Morphology

- Words of a language have similarities between them and regularities within their structure
- A single word can be viewed as made up of smaller units, called **morphemes**
- **free morphemes**: act as words in isolation (e.g. *think*, *permanent*, *local*)
- **bound morphemes**: operate only as parts of other words (e.g. *ize*, *ing*, *re*)
- In English, bound morphemes are usually **affixes**; **suffixes** at the end of the word; **prefixes** at the front

Morphology

- The morpheme that forms the central meaning unit in a word is called the **stem**
- A word is made up of one or more morphemes:
  - *think* stem *think*
  - *rethink* prefix *re*, stem *think*
  - *localize* stem *local*, suffix *ize*
  - *delexicalize* prefix *de*, stem *lexical*, suffix *ize*

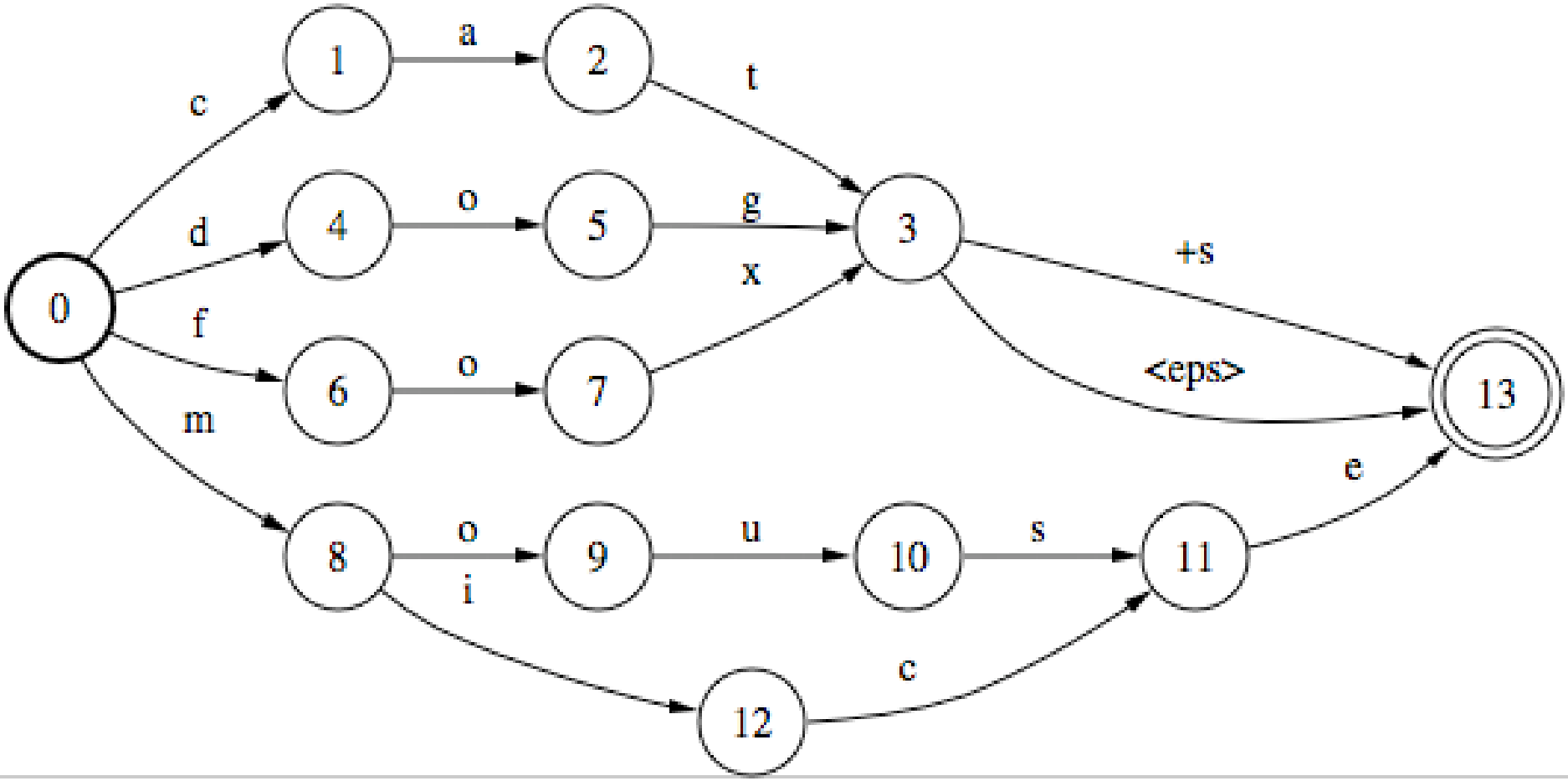
Inflectional Morphology

- Assume we want to represent the fact that English nouns form plurals (to avoid duplication in dictionaries)
- Very simple type of grammar development
- The plural:

  cat+N+PL := cats dog+N+PL := dogs, etc.
- Exceptions:

  fox+N+PL := foxes mouse+N+PL := mice
- Exercise: draw an FSM to represent the 4 words above. Condition: only use a single transition for the plural suffix *+s*

FSM for simple morphology



FSM for simple morphology

- Problem: represents *foxes* as *foxs*
- Not a pure irregular case like *mouse*/*mice*
- e.g. *pass*/*passes*, *fax*/*faxes*, *suffix*/*suffixes*
- We will revisit this problem with a model more powerful than FSMs

FSM for simple morphology

- Lessons from grammar development:
  - Many regular cases (captured by a simple rule)
  - Some exceptional cases (*mice*) that are irregular and have no simple generative rule
  - Some cases that appear irregular but are rules that apply in the right context
- These observations are pervasive in grammar development (for other modules of natural language)

Derivational morphology

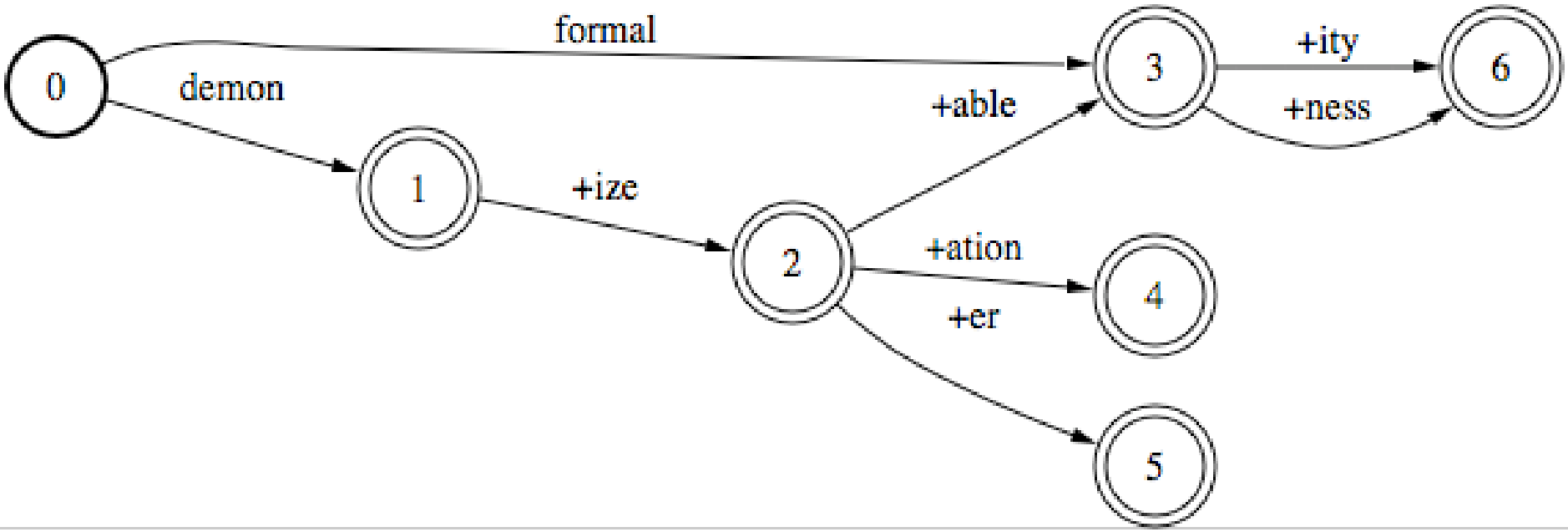
- Some suffixes can change the part of speech for a word, e.g. demon/N demonize/V
- We will write demonize/V as demon+ize/V to emphasize the suffix addition
- Notation, N: noun V: verb A: adjective
- Draw an NFA for the following data:

  demon/N demon+ize/V demon+ize+ation/N

  demon+ize+able/A demon+ize+er/N

  formal/N formal+ity/N formal+ness/N

Derivational morphology



Derivational morphology

- Now modify the original NFA to accept the following additional strings:

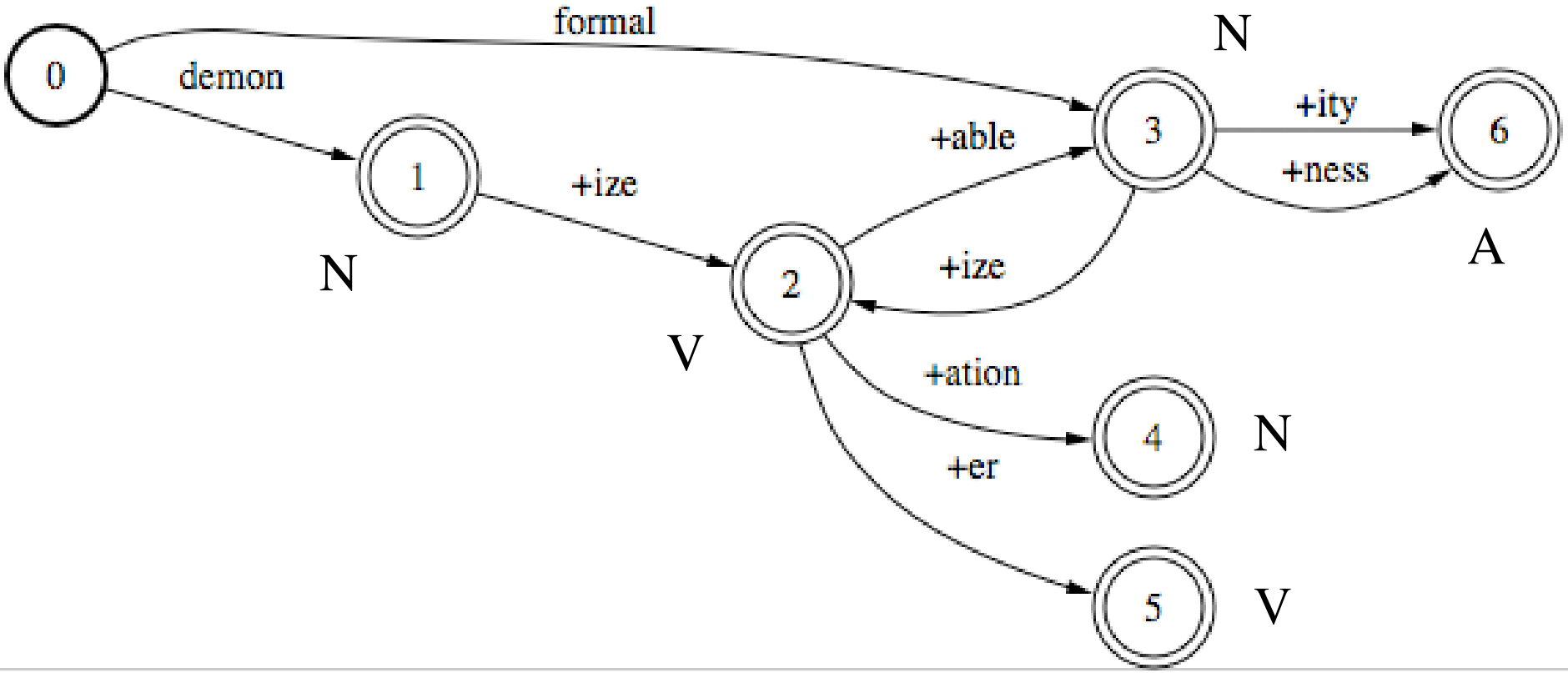
  formal+ize/V formal+ize+ation/N

  formal+ize+able/A formal+ize+er/N

  demon+ize+able+ity/N

- Note that we assume that *demon+ize+able+ity* is mapped to *demonizability*

Derivational morphology



Capturing the derivation of a part of speech:
Map states to symbols

Morphology and lexicons

- Constructing a lexicon can benefit from the morphological analysis of words
- Especially in languages with productive morphology. For example, in Turkish:

uygarlaştıramadıklarımızdanmışsınızcasına

uygar + laş + tır + ama + dık + lar

civilized + "become" + "to cause" + "not able" + ppart + pl

+ ımız + dan + mış + sınız + casına

+ 1stpers possessive + "from/among" + past + 2nd person pl + adverbial

(behaving) as if you are among those whom we could not cause to become civilized

Morphology and lexicons

- Morphology is useful for lexicons:
  - Smaller size for the lexicons: various forms are computed by applying a rule rather than storing all possible forms
  - Ease of entering data: only the stem and category needs to be added and all possible forms are computed
  - New word forms (perhaps previously unknown to the lexicon builder) are produced. e.g. demonizability
  - Simpler and faster lookup (when using finite-state methods)

Summary

- Some aspects of morphology can be captured in a FSM, e.g. cat+N+PL := cats
- Other aspects are not elegantly captured. We have to create a sub-class for words like fox+N+PL := foxes and treat them differently
- Algorithms for FSMs can be used to recognize/generate/store these simple linguistic forms