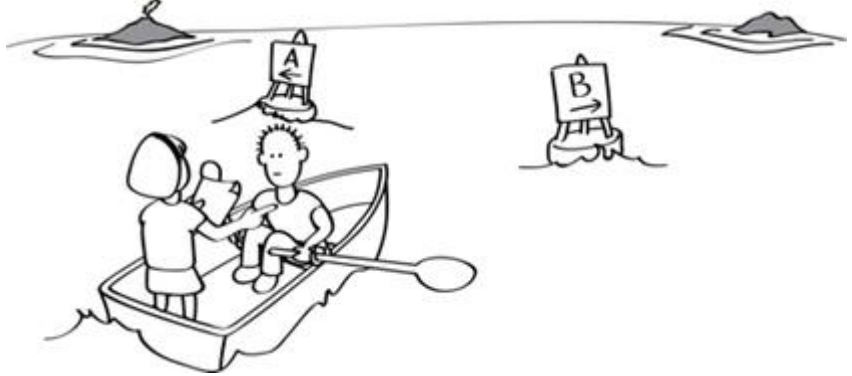


CPE 372/641 Natural Language Processing

Finite State Automata (FSA)

Asst. Prof. Dr. Nuttanart M. Facundes

(most contents from Mark Granroth-Wilding slides, University of Helsinki)

REs revisited

Pattern	*Matches*
`/final*/`	we finally restored it,
`/final+/`	we finally restored it,
`/radio./`	conventional radioese, I repeated
`/.(it|ah)(-.(it|ah))*/`	'Dah-dit-dah-dit dah-dah-dit-dah.

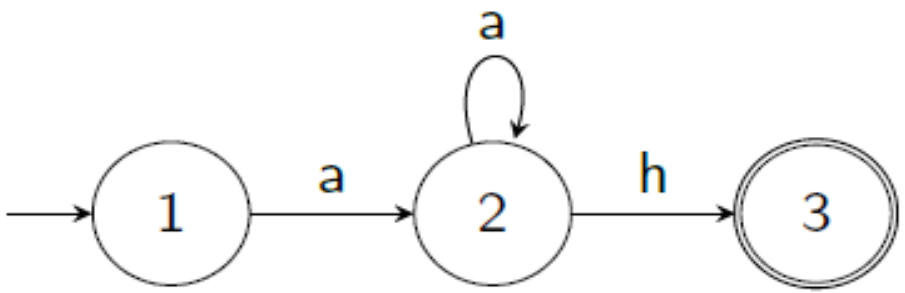
- Regular expression defines a regular language.
- Set of strings accepted by regex

  $L(r) = \{s \mid accept(r,s)\}$

- Language is regular iff ∃ regex for it

Finite State Automata (FSA)

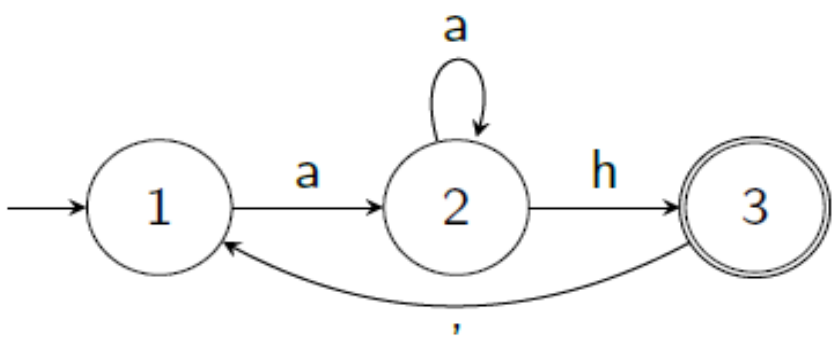
- Another string-testing formalism: **finite-state automaton** (FSA)
- Process string by transitioning between states



{ '*ah*', '*aah*', '*aaah*', ... }

- Equivalent regex: `/a+h/`

ANOTHER FSA

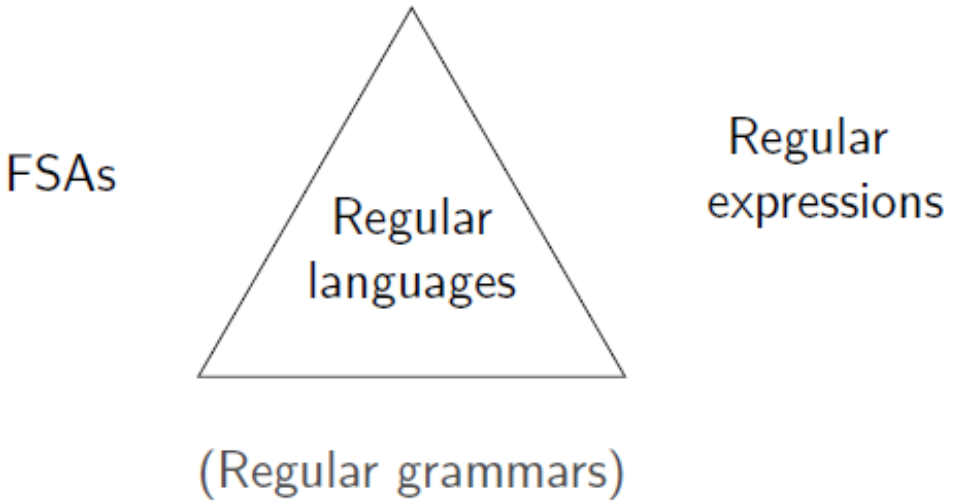


{ '*ah, ah*', '*ah, aah*', '*aaaaah, ah, aah*', ... }

- Equivalent regex: `/a+h(, a+h)*/`

FSA and RE

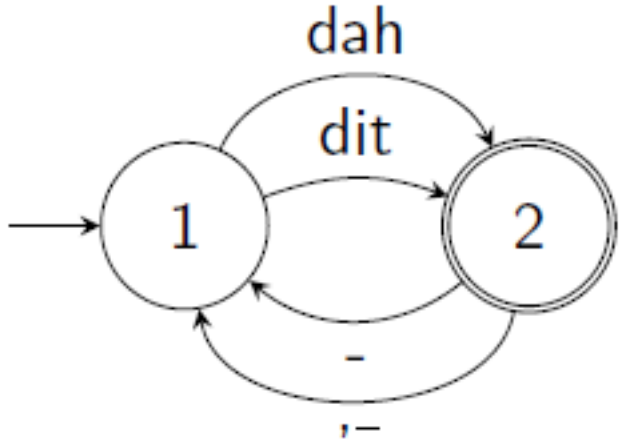
- Acceptance by FSA $\equiv$ acceptance by regex
- Every FSA has equivalent regex



Finite State Transducers (FST)

- Extend FSA to *output* something: not just YES/NO
- Each accepting edge also outputs
- Finite-state **transducer**
- Same strings/language as FSA
- Output may be:
  - translation
  - analysis
  - spelling transformation
  - ...

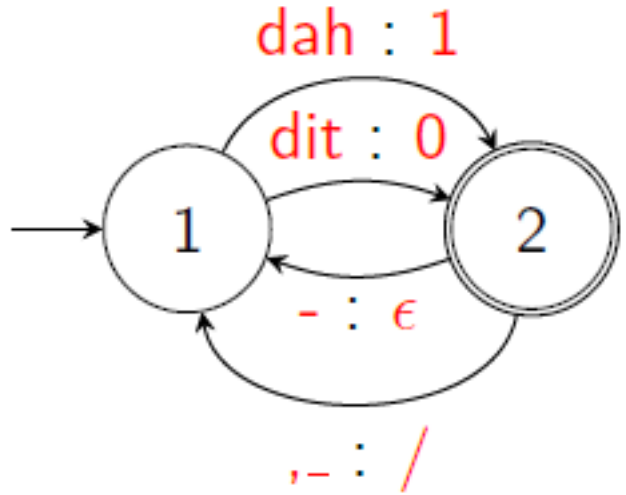
FSA → FST



{ '*dit-dah-dah, dah-dit*', ... }

```
/(dit|dah)(-(dit|dah))*(, (dit|dah)(-(dit|dah))*)*/
```

FSA → FST



- Each state: *input : output*
- Translates

  dit-dah-dah, dah-dit, dit ⇒ 0-1-1/1-0/1

- Common use in NLP: analysis of internal word structure
  → *morphology*

Example

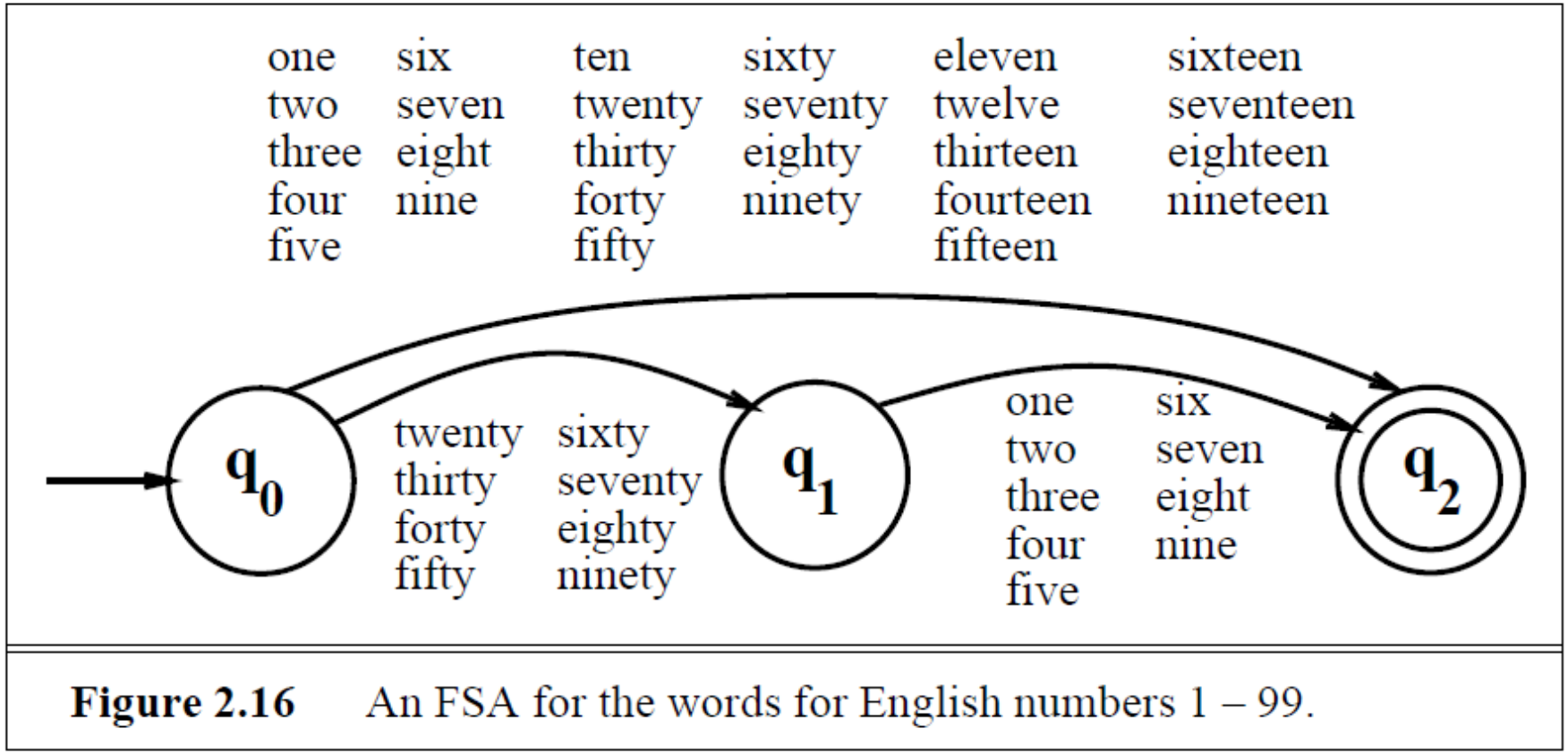


Figure 2.16 An FSA for the words for English numbers 1 – 99.