

Finite State Machines

CS114 Lab 2

Kenneth Lai

January 24, 2020

Finite State Machines

- ▶ Computers with almost no memory

Finite State Machines

- ▶ Computers with almost no memory
- ▶ Useful for morphology (but not syntax)

Finite State Machines

- ▶ Automata
 - ▶ Output: just accept or reject

Finite State Machines

- ▶ Automata
 - ▶ Output: just accept or reject
- ▶ Transducers
 - ▶ Output: an output string (or anything you want)

Finite State Machines

- ▶ Input: a string (containing symbols from some alphabet)

Finite State Machines

- ▶ Input: a string (containing symbols from some alphabet)
 - ▶ Read one symbol at a time from left to right

Finite State Machines

- ▶ Input: a string (containing symbols from some alphabet)
 - ▶ Read one symbol at a time from left to right
- ▶ Start in the start (initial) state

Finite State Machines

- ▶ Input: a string (containing symbols from some alphabet)
 - ▶ Read one symbol at a time from left to right
- ▶ Start in the start (initial) state
- ▶ Transition to next state according to current state and read symbol

Finite State Machines

- ▶ If you get to the end of the input and you are in:

Finite State Machines

- ▶ If you get to the end of the input and you are in:
 - ▶ Accept (final) state: return “accept”/output string

Finite State Machines

- ▶ If you get to the end of the input and you are in:
 - ▶ Accept (final) state: return “accept”/output string
 - ▶ Other state: return “reject”/do not output string

Finite State Machines

- ▶ Determinism—exactly one transition for each combination of state/symbol

Finite State Machines

- ▶ Determinism—exactly one transition for each combination of state/symbol
- ▶ Nondeterminism—zero or more transitions for each combination of state/symbol

Finite State Machines

- ▶ For transducers, distinguish between nondeterministic transducers and nondeterministic transductions

Finite State Machines

- ▶ For transducers, distinguish between nondeterministic transducers and nondeterministic transductions
 - ▶ Transducer—as before

Finite State Machines

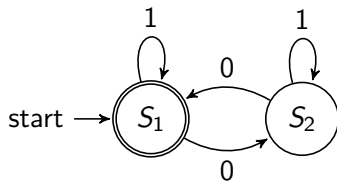
- ▶ For transducers, distinguish between nondeterministic transducers and nondeterministic transductions
 - ▶ Transducer—as before
 - ▶ Transduction—more than one possible output string for any input string

Finite State Machines

- ▶ For transducers, distinguish between nondeterministic transducers and nondeterministic transductions
 - ▶ Transducer—as before
 - ▶ Transduction—more than one possible output string for any input string
 - ▶ For PA, make sure all transductions are deterministic

Finite State Machines

- State diagram



Finite State Machines

- ▶ Language recognized by a finite state machine
 - ▶ Set of input strings for which the machine returns “accept” /output string

Finite State Machines

- ▶ Language recognized by a finite state machine
 - ▶ Set of input strings for which the machine returns “accept” /output string
- ▶ A language is regular if and only if it is recognized by a finite state machine