UCF | College of Engineering and Computer Science

UNIVERSITY OF CENTRAL FLORIDA

Finite Automata

COP-3402 Systems Software
Paul Gazzillo

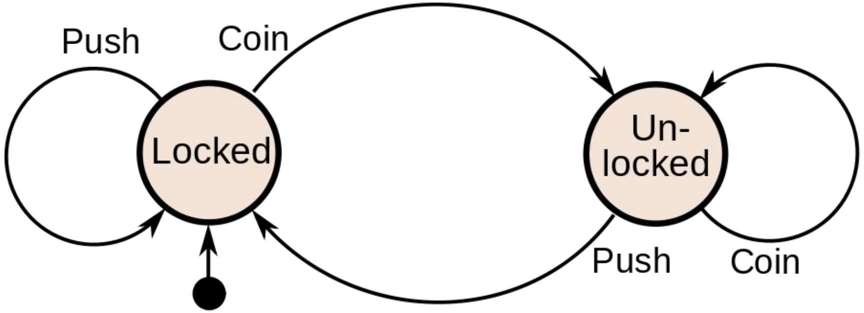
UCF

Recognizing Regular Expressions

- Given an expression (ab|d)*
- Can hand-implement a specific regular expression
- We can automate their generation
- Regular languages correspond to computation

Finite Automata

- States and transitions
- Example: turnstile
- Turnstile states: locked and unlocked
- Turnstile transitions: "push" and "coin"
- Transitions move from state to state
  - E.g., coin causes an unlock, push returns to locked

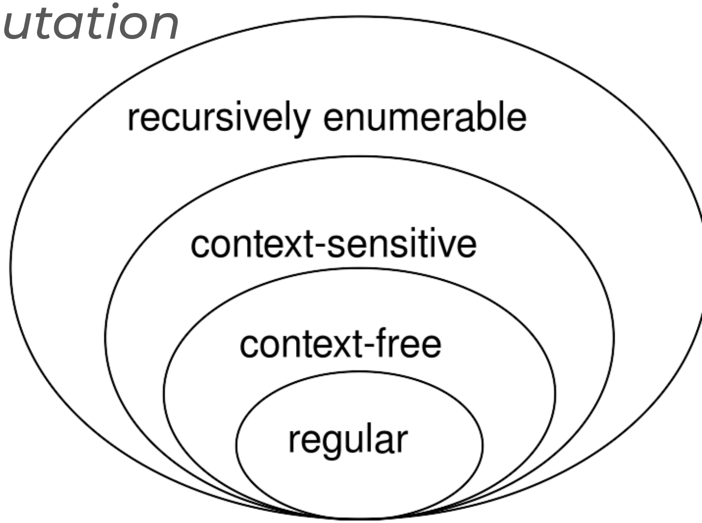


Finite Automata Are a Model of Computation

- Corresponds to regular languages
  - Any regular expression can be recognized by a finite automaton and vice versa
- Also called
  - Finite state machines
  - Finite state automata
- Applications
  - Vending machines: count coins
  - Elevators: sequence of stops
  - Traffic lights: order of changes
  - Combination lock: numbers in correct order

Chomsky Hierarchy of Languages

- Equivalence between *language* and *computation*
- Regular languages
  - Finite automata, regular expression
- Context-free languages
  - Syntactic structures
  - Parsers, push-down automata
- Context-sensitive
  - Syntax depends on surrounding context
  - (Fancier) parsers, non-deterministic push-down automata
- Recursively enumerable
  - Recognized by a Turing machine



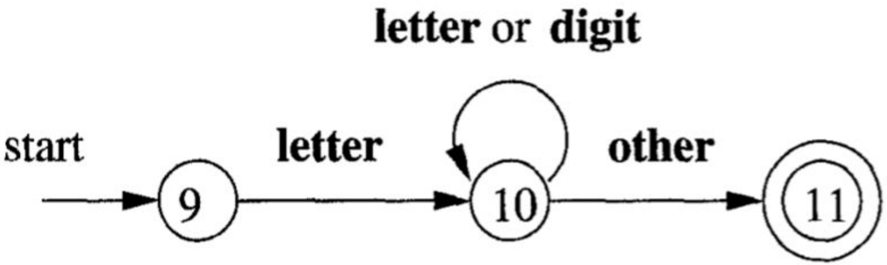
https://en.wikipedia.org/wiki/Chomsky_hierarchy#/media/File:Chomsky-hierarchy.svg
J. Finkelstein
https://creativecommons.org/licenses/by-sa/3.0/

Finite Automata Define State Transitions

- Defines states and transitions between them
  - Keep reading characters until the end of input
- Definition of automata: like regular expressions
  - Finite alphabet of symbols (like regular languages):
  - Finite set of states:
  - State transition function:
  - An initial state and a set of final states:
- Graphical representation with state diagrams
  - States: circles
  - Transitions: labeled arrows
  - Starting state: in-arrow
  - Accepting states: doubled circles

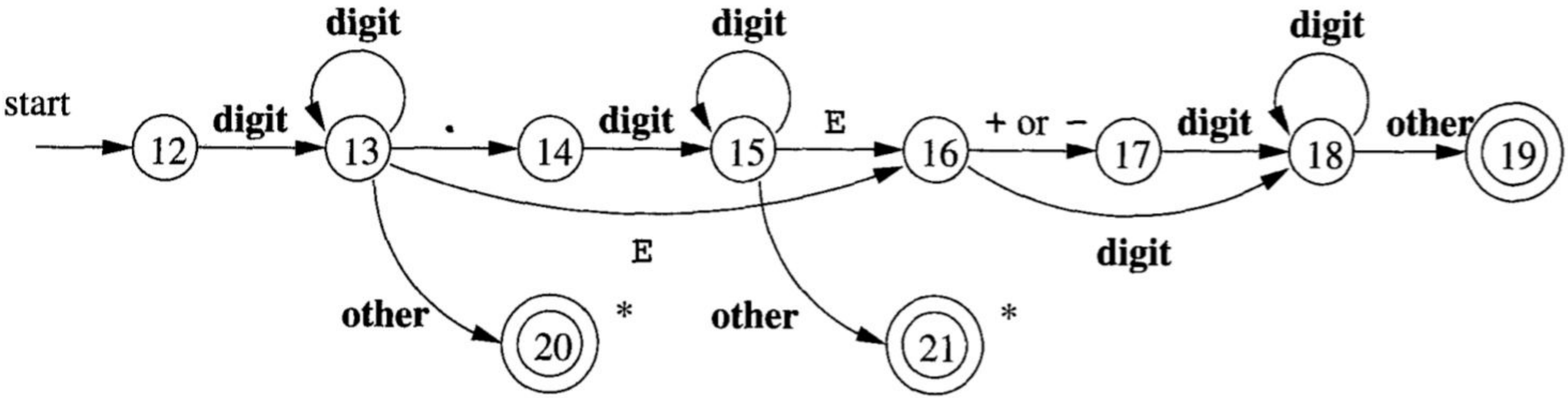
Example of Automaton for Identifiers

- Letters and digits are character classes
  - Makes drawing the diagram easier
- Other is the class of characters besides letters and digits
  - Represents the decision based on a lookahead
  - Matches longest identifier sequence



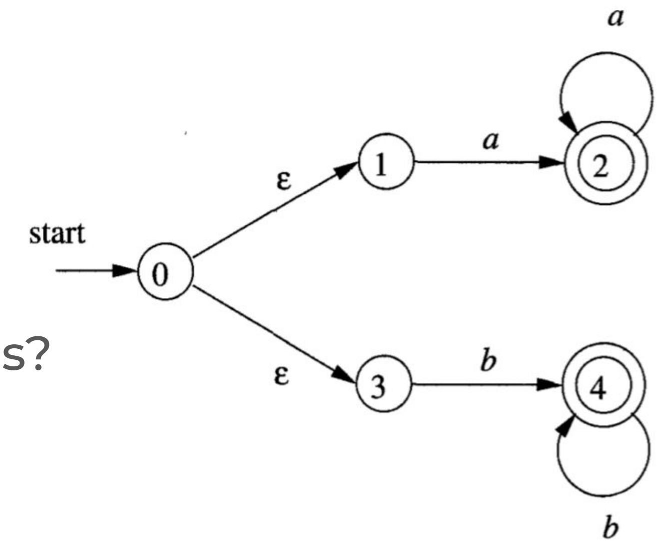
More Complex Automaton for Numbers

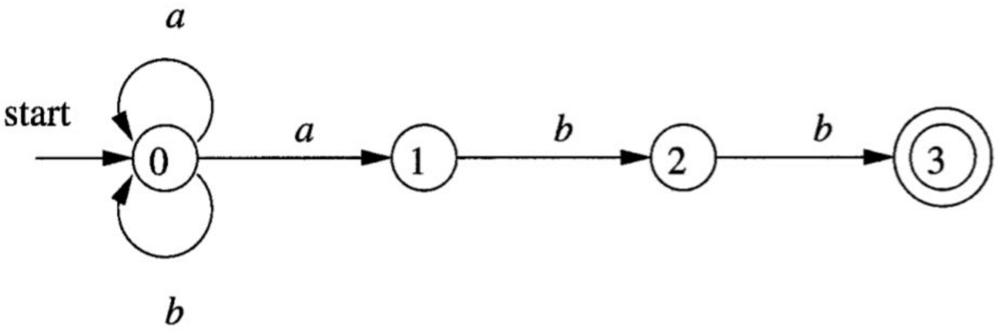
- What kinds of numbers can be represented?
- What is an equivalent regular expression?



Nondeterministic vs. Deterministic

- Deterministic: one state at-a-time
- Nondeterministic: multiple states at once
  - Same symbol, multiple transitions
  - Epsilon transitions (epsilon is empty string)
- What regular expressions are these figures?





Figures 3.24 and 3.26
from the Dragon Book

Demo: Example Finite Automata

L(L|D)*

(a|b)*abb

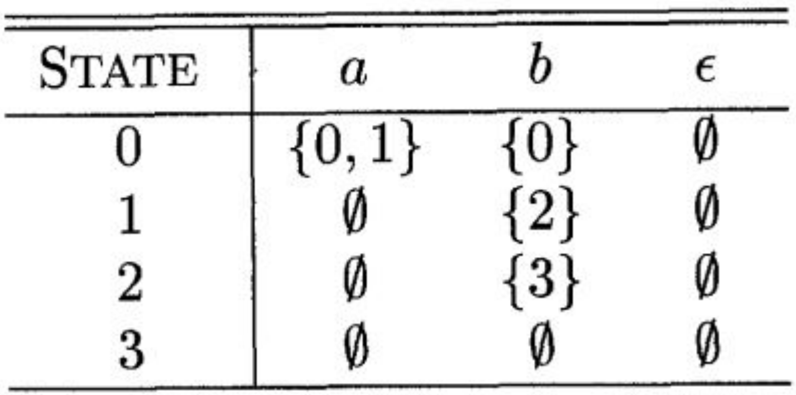
aa*|bb*

Properties of Finite Automata

- Any regular expression can be represented with an FA
- Any NFA can be represented with a DFA
- Many regular expressions or FAs can represent the same language
- There is a minimal DFA for given language (fewest states)

Finite Automata with Transition Tables

- Instead of state diagram, can use a table
- Convenient representation for a program
- DFAs vs NFAs
    - use the DFAs and NFAs from the demo



STATE	a	b	ϵ
0	$\{0, 1\}$	$\{0\}$	$\emptyset$
1	$\emptyset$	$\{2\}$	$\emptyset$
2	$\emptyset$	$\{3\}$	$\emptyset$
3	$\emptyset$	$\emptyset$	$\emptyset$

Demo: Example Transition Tables

L(L|D)*

(a|b)*abb

aa*|bb*

The Power of State Machines

- Solving Pokemon Blue with a single, huge regular expression https://www.youtube.com/watch?v=Q2g9d29UIzk&feature=youtu.be&t=253
- Other games
  - State machines in Doom https://www.moddb.com/games/doom/tutorials/doom-source-code-tutorial-5
  - Using state machines to control behavior and animations https://gamedevacademy.org/how-to-use-state-machines-to-control-behavior-and-animations-in-phaser/

Conclusion

- Finite state machines are an abstract model of computation
- Regular expressions describe string patterns
- *Both* represent regular languages
  - Each regular expression has an equivalent finite automata (and vice versa)
- Automata have many applications in computing and engineering
  - Games
  - Elevators
  - Compilers
- Finite automata can be implemented in code
  - Conversion from regular expression to automata to code can be fully automated