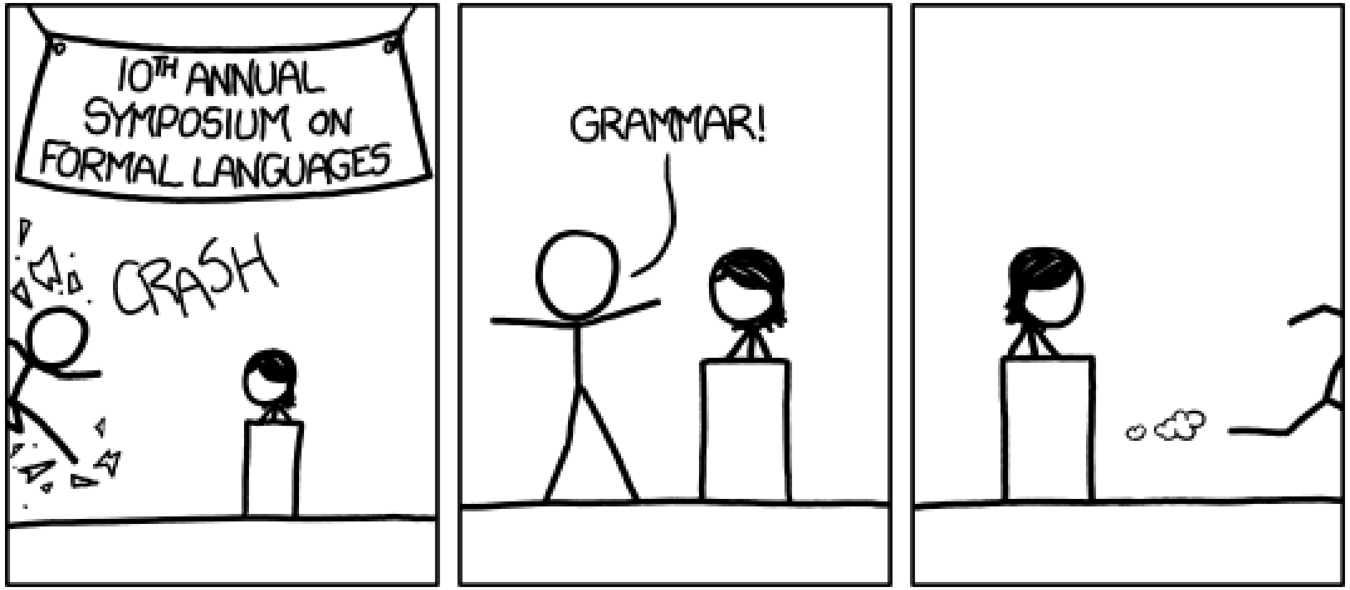



# Context Free Grammars, Deterministic Finite Automata

CSE 311: Foundations of Computing I
Lecture 20

# Announcements

- HW6 due Wednesday at 11:59pm on Gradescope.
- Midterm Corrections due Wednesday at 11:59pm on Gradescope.
  - $\text{midterm\_grade} = 0.75 \cdot \text{original\_grade} + 0.25 \cdot \max(\text{corrected\_grade}, \text{original\_grade})$
  - No Late Days permitted
  - More details posted on Assignments page

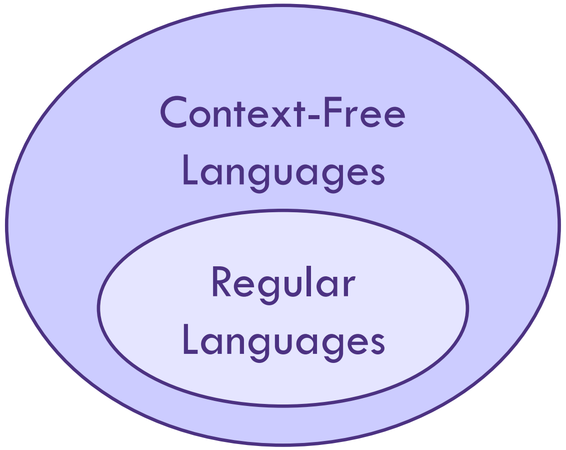
# Recall: Languages

- A **language** is a set of strings.
- A computer is said to **recognize** a language if it can distinguish which strings are in a language vs. which are not.
- Regular Languages: Languages that can be specified by a Regular Expression.
- Context Free Languages: Languages that can be generated by a Context Free Grammar

# Context Free Grammars

# Context-Free Languages

- Many languages are irregular, e.g. binary palindromes
- Context-Free Languages are a strictly larger class of languages




- Context-Free Languages are generated by Context-Free Grammars
  (just like Regular Languages are specified by Regular Expressions)

# Example

$S \rightarrow Ab \mid c$

$A \rightarrow Aa \mid \varepsilon$

Generates the same language as the Regular Expression $(c \cup a^* b)$

# Example

$S \to 0S \mid S1 \mid \varepsilon$

The set of all binary strings with any number of 0s followed by any number of 1s

# Example

The set of all binary palindromes.

$S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1 \mid \varepsilon$

# Example

CFG for the language $\{0^n 1^n : n \geq 0\}$

$S \rightarrow 0S1 \mid \varepsilon$

# Example

CFG for the language $\{0^n1^n23 : n \geq 0\}$

$S \rightarrow A23$

$A \rightarrow 0A1 \mid \varepsilon$

# Exercises

CFG for the set of binary strings with the same number of 0s as 1s.

$S \rightarrow SS \mid 0S1 \mid 1S0 \mid \varepsilon$

CFG for the set of balanced parentheses. E.g. ((())())

$S \rightarrow SS \mid (S) \mid \varepsilon$

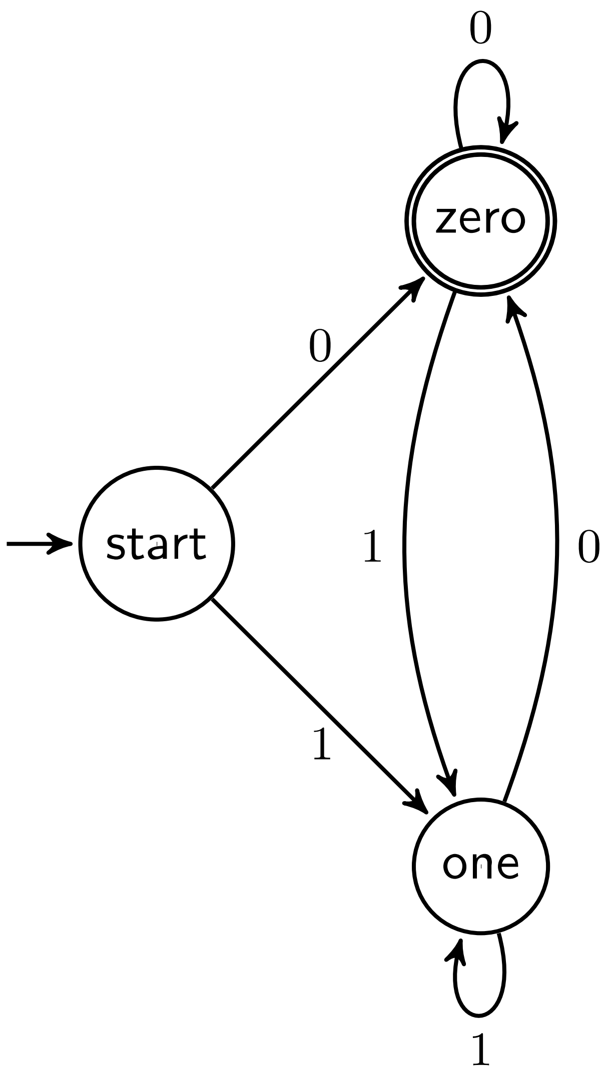
# Deterministic Finite Automata

# Model of Computation

- We want a mathematical model for a computer
- We'll focus on computers whose input is a string, and output is true/false on whether that string is in a particular language
- The first model we'll study is called a Deterministic Finite Automata (DFA)

# Deterministic Finite Automata (DFA)

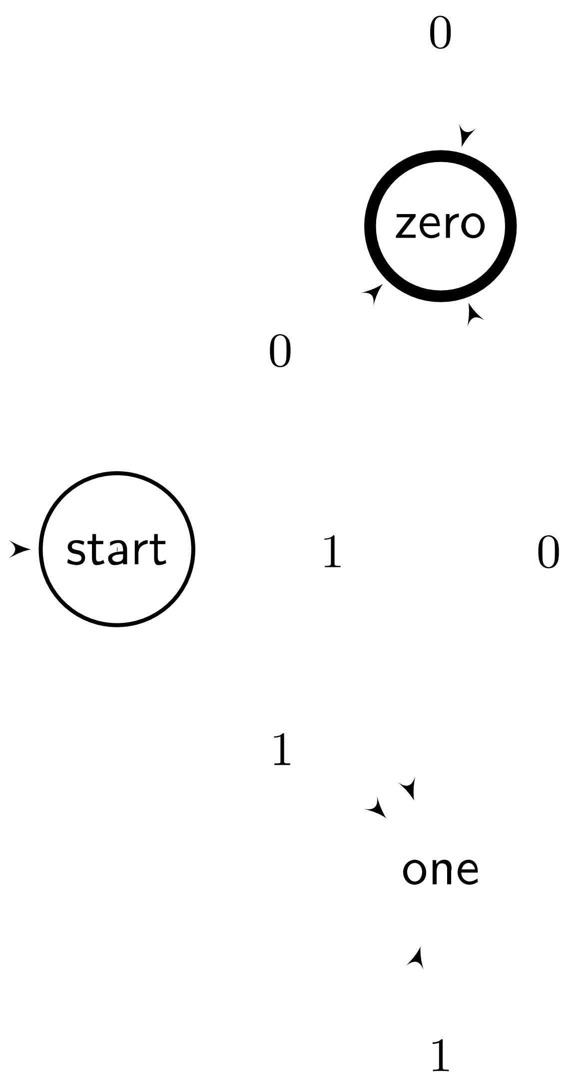
- Our machine will have a finite number of states.
- It will take a string as input.
- It will read one character at a time and update “its state”.
- When it reads the character it follows the arrow labeled with that character to its next state.
- Start at the “start state” (unlabeled, incoming arrow).
- After you’ve read the last character, accept the string if and only if you’re in an “accept state” (double circle or bolded).



# Deterministic Finite Automata (DFA)

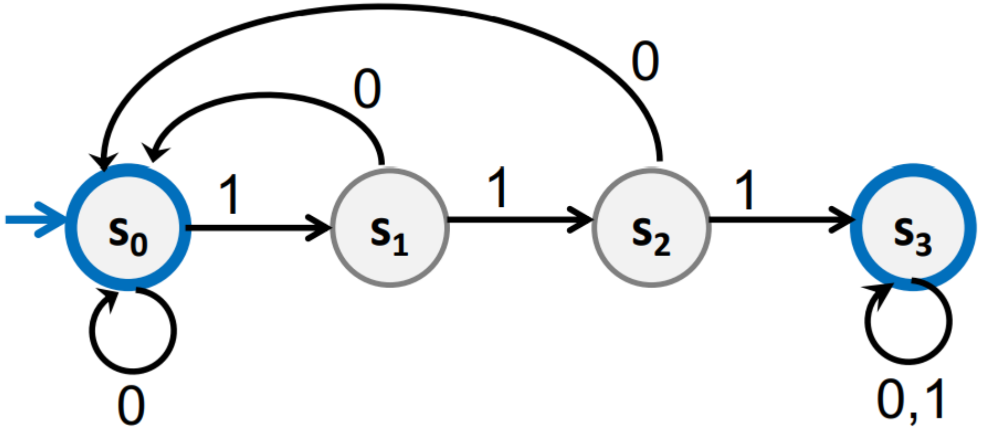
What language does this DFA recognize?

All binary strings that end in a **0**.



# Example

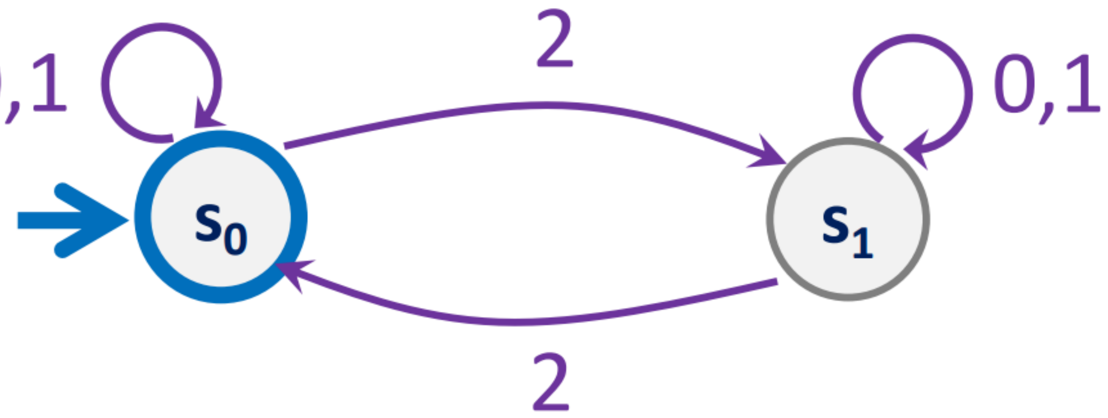
What language does this DFA recognize?




The set of all binary strings that contain 111 or end in $\mathbf{0}$.

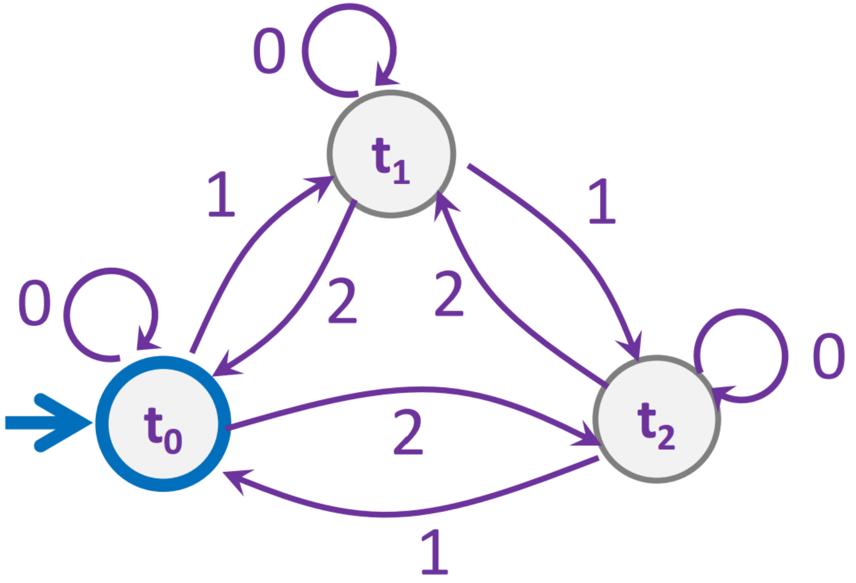
# Example

Design a DFA that accepts the language of strings over $\{0, 1, 2\}$ with an even number of 2's.



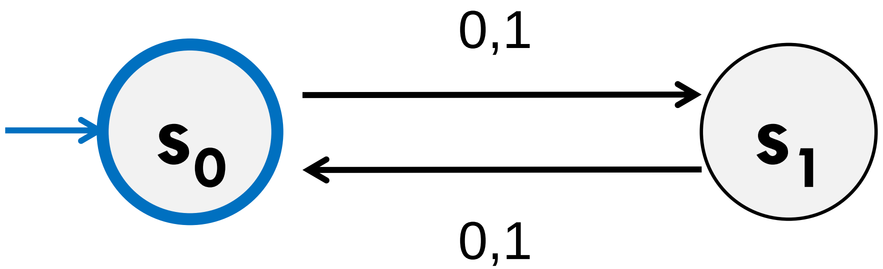
# Example

Design a DFA that accepts the language of strings over $\{0, 1, 2\}$ where the sum of digits mod $3$ is $0$.



# Exercises

DFA for all binary strings of even length.




DFA for the set of binary strings with a 1 in the 3rd position from the start.