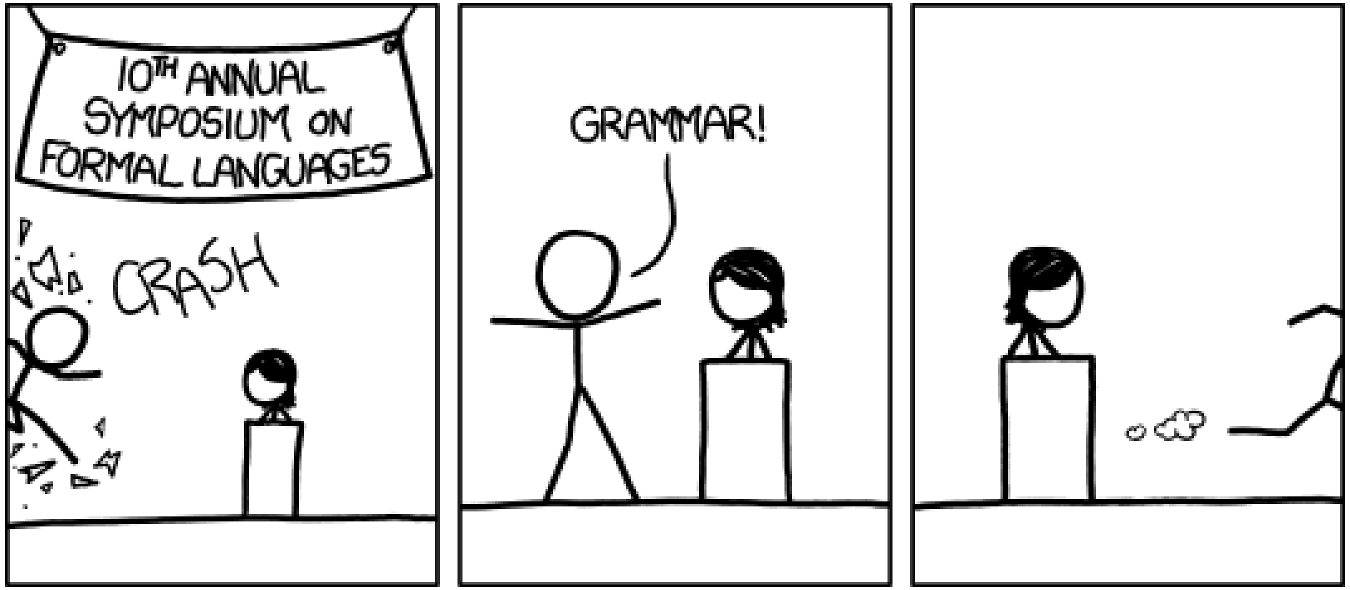


Context Free Grammars, Deterministic Finite Automata

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

Binary Palindromes | $(0 \cup 1)^* 1$

X 111101111

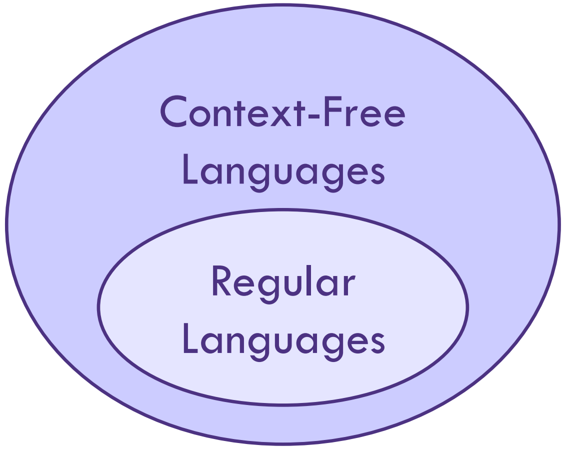
- A **language** is a set of strings. L: Set of binary strings that end in 1
- A computer is said to **recognize** a language if it can distinguish which strings are in a language vs. which are not. $001 \in L$ $100 \notin L$
- Regular Languages: Languages that can be specified by a Regular Expression.
- Context Free Languages: Languages that can be generated by a Context Free Grammar

Context Free Grammars

Context-Free Languages

binary strings w/ same # of 0s & 1s

- 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

$\{c, b, ab, aab, \ldots\}$

$S \to Ab \mid c$

$A \to Aa \mid \varepsilon$

"generating"

$S \Rightarrow Ab \Rightarrow Aab \Rightarrow Aaab$

$\Rightarrow aab$

$\boxed{c \cup a^*b}$

- Production Rules (2)

  $S \to Ab \mid c$

  $A \to Aa \mid \varepsilon$

- Nonterminals (S, A, ...) (2)

  S

  A

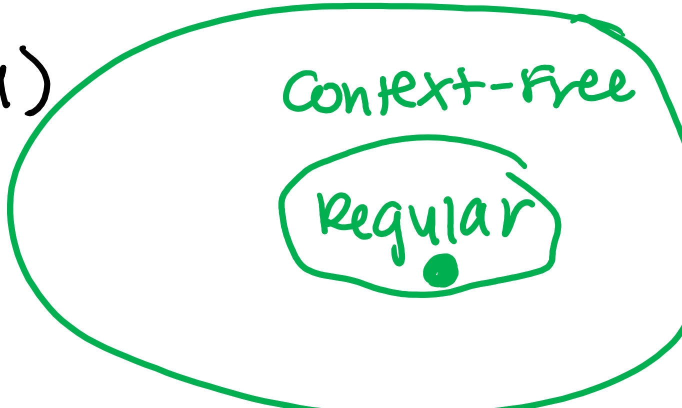
  "variables"

  - Specific nonterminal that is the start nonterminal (S)

- Terminal (4)

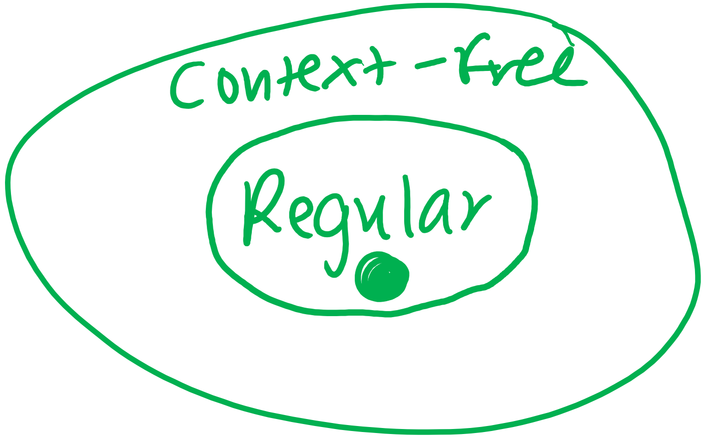
  a c

  b ε



Example

$0^* 1^*$



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

$S \Rightarrow 0S \Rightarrow 0S1 \Rightarrow 0S11 \Rightarrow 011$

Binary strings with 0s at the front followed by 1s at the end.

All binary strings with any # of 0s followed by any # of 1s

Example

of 0 & 1

read the same forwards & backwards

The set of all binary palindromes.

① 11011 odd length

② 010010 even length

ε, 0, 1

even odd

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

$$S \Rightarrow 0S0 \Rightarrow 01S10 \Rightarrow 010S010 \Rightarrow 010010$$

Example

n copies of 0 n copies of 1

CFG for the language $\{0^n 1^n : n \geq 0\}$ $\{\varepsilon, 01, 0011, 000111, \ldots\}$

$$\boxed{S \rightarrow 0S1 \mid \varepsilon}$$

0011 ✓

$$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 0011$$

Example

Tip: Break up into more production rules

CFG for the language $\{0^n 1^n 23 : n \geq 0\}$

$\{23, 0123, 001123, \ldots\}$

✗ $S \to 0S1 \mid 23$

$S \Rightarrow 0S1 \Rightarrow 0231$

✓ $\boxed{\begin{array}{l} S \to A23 \\ A \to 0A1 \mid \varepsilon \end{array}}$

0^n1^n23

0^n1^n

$S \Rightarrow A23 \Rightarrow 0A123$

$\Rightarrow 00A1123$

$\Rightarrow 001123$

Exercises

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

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

Deterministic Finite Automata

(DFA)

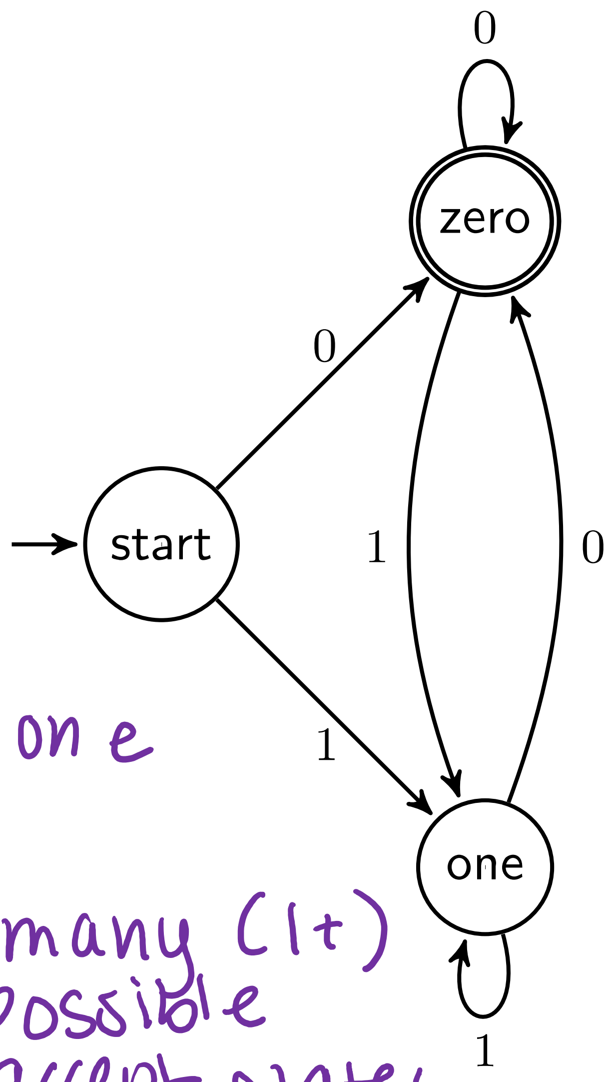
Model of Computation

- 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 the input string is in a particular language
- The first model we'll study is called a Deterministic Finite Automata (DFA)

Deterministic Finite Automata (DFA)

~~1~~ ~~0~~ ~~1~~ ~~1~~ rejected
4

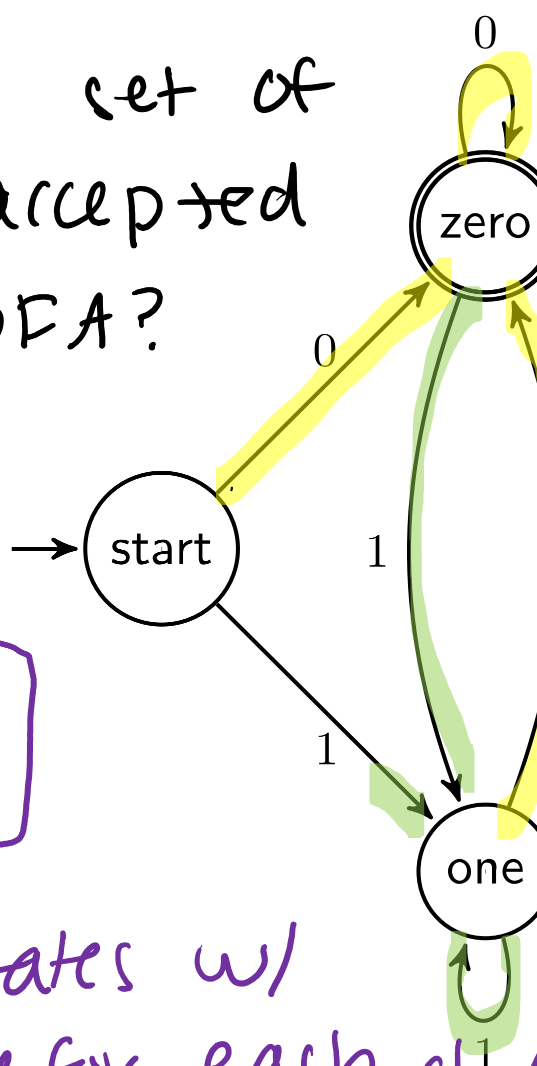
- Our machine will have a finite number of states.
- It will take a string as input. 1011
- It will read one character at a time and update "its state".
- When it reads the character it follows the arrow labeled with that character to its next state.
- Start at the "start state" (unlabeled, incoming arrow). only one
- After you've read the last character, accept the string if and only if you're in an "accept state" (double circle or bolded).



many (1+) possible accept states

Deterministic Finite Automata (DFA)

What language does this DFA recognize? i.e which set of strings are accepted by the DFA?



1011 - rejected

100 - accepted

All binary strings that end in 0.

· states

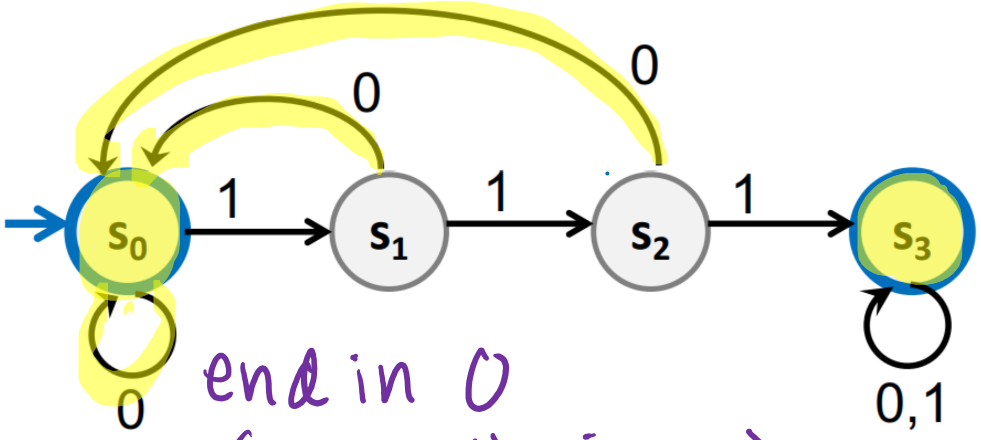
accept reject

· edges between states w/ labels (one for each char)

Example

What language does this DFA recognize?

inputs: binary strings



end in 0
(& aren't in s_3)

S_0, S_3 are accept states

S_1, S_2 are reject states

all binary strings that contain 111.

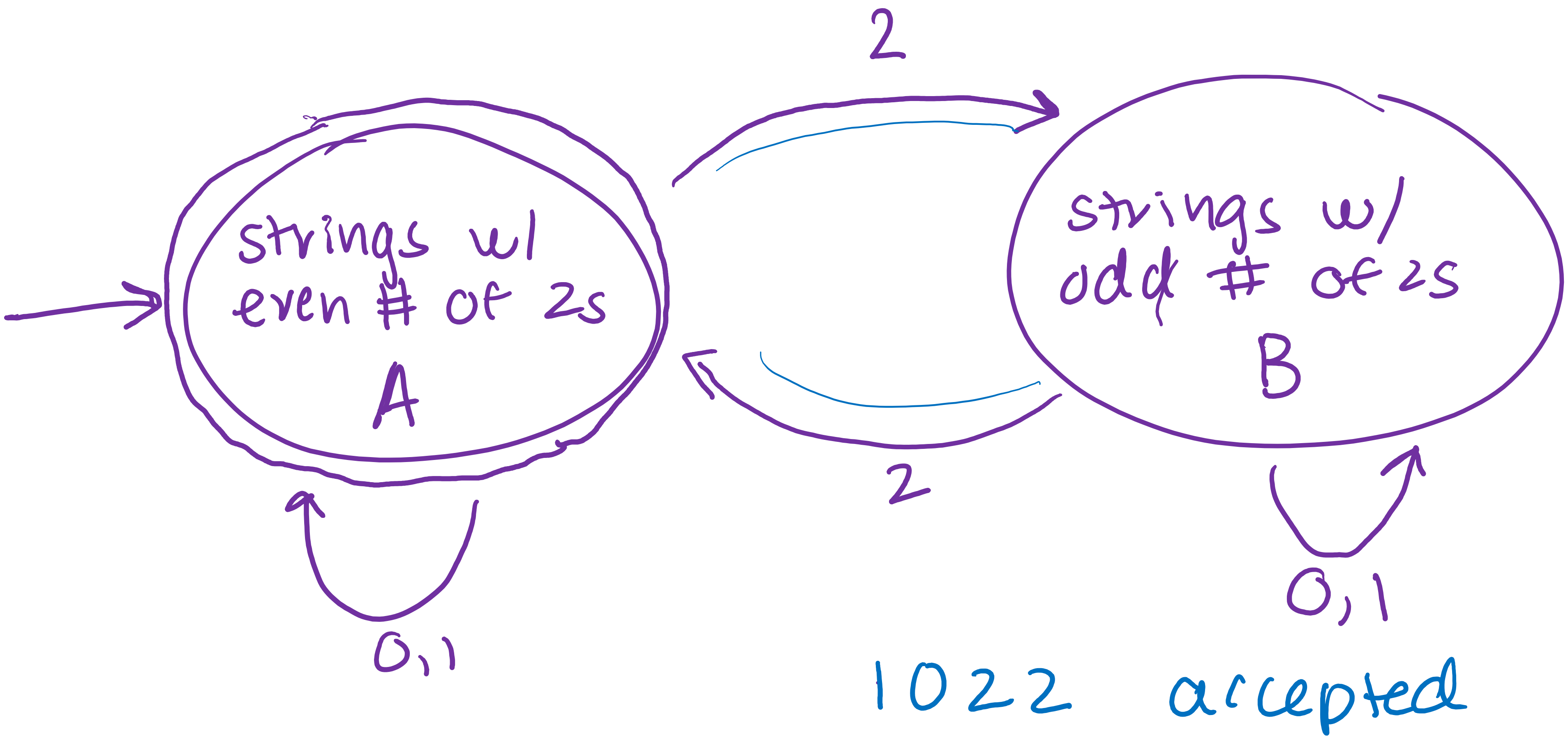
The set of all binary strings that contain 111 or end in 0.

Example

3. Connect states w/ edges

1. What states?
2. Which is the start state?
figure out where ε goes

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 3^{rd} position from the start.