Join WACM

HMC chapter of acm-w

- Company sponsored tech talks and events
- Resume Workshops / Interview Prep
- Dinners with cool profs (like the one standing in front of you!)
- Meet other students in CS!

Students of all gender identities are welcome!

Join by filling out a quick form at tinyurl.com/wacm-5c

Visit us at the HMC club fair and the 5c turf dinner.

How would you tell if a binary number is even or odd?

Full name

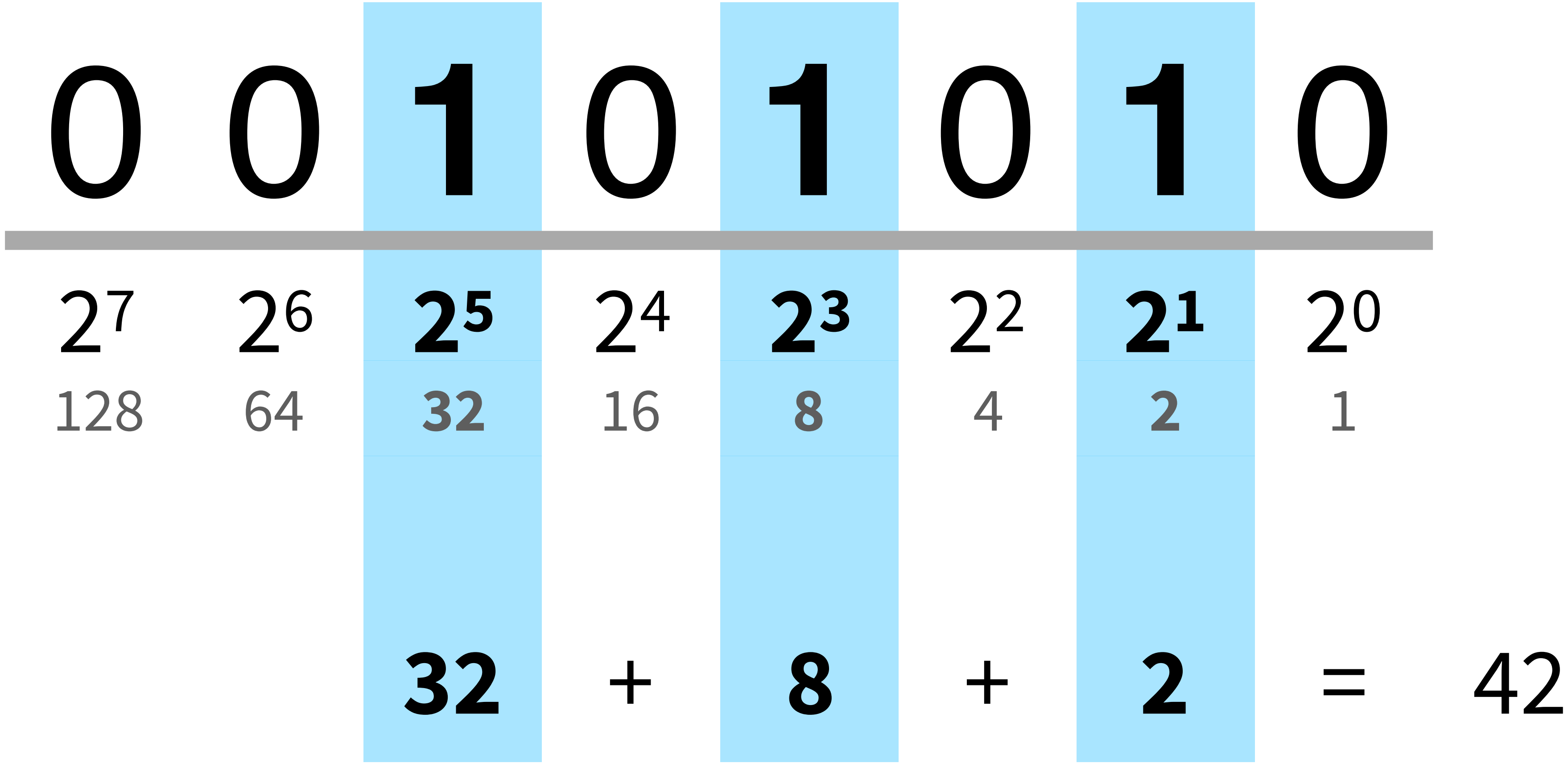
R. 9/6

Quick binary refresher

A **bit** is a binary digit: 0 or 1.

A **bitstring** is a sequence of zero or more bits.

We can assign a **numeric value** to non-empty bitstrings.



0	0	**1**	0	**1**	0	**1**	0
2^7	2^6	$\mathbf{2^5}$	2^4	$\mathbf{2^3}$	2^2	$\mathbf{2^1}$	2^0
128	64	**32**	16	**8**	4	**2**	1

32 + **8** + **2** = 42

Motivating questions for this week and next

What kinds of problems

can computers solve?

What do we mean by “problem”?

How do we describe the problem?

How do we recognize a solution?

we need a
"formal model"

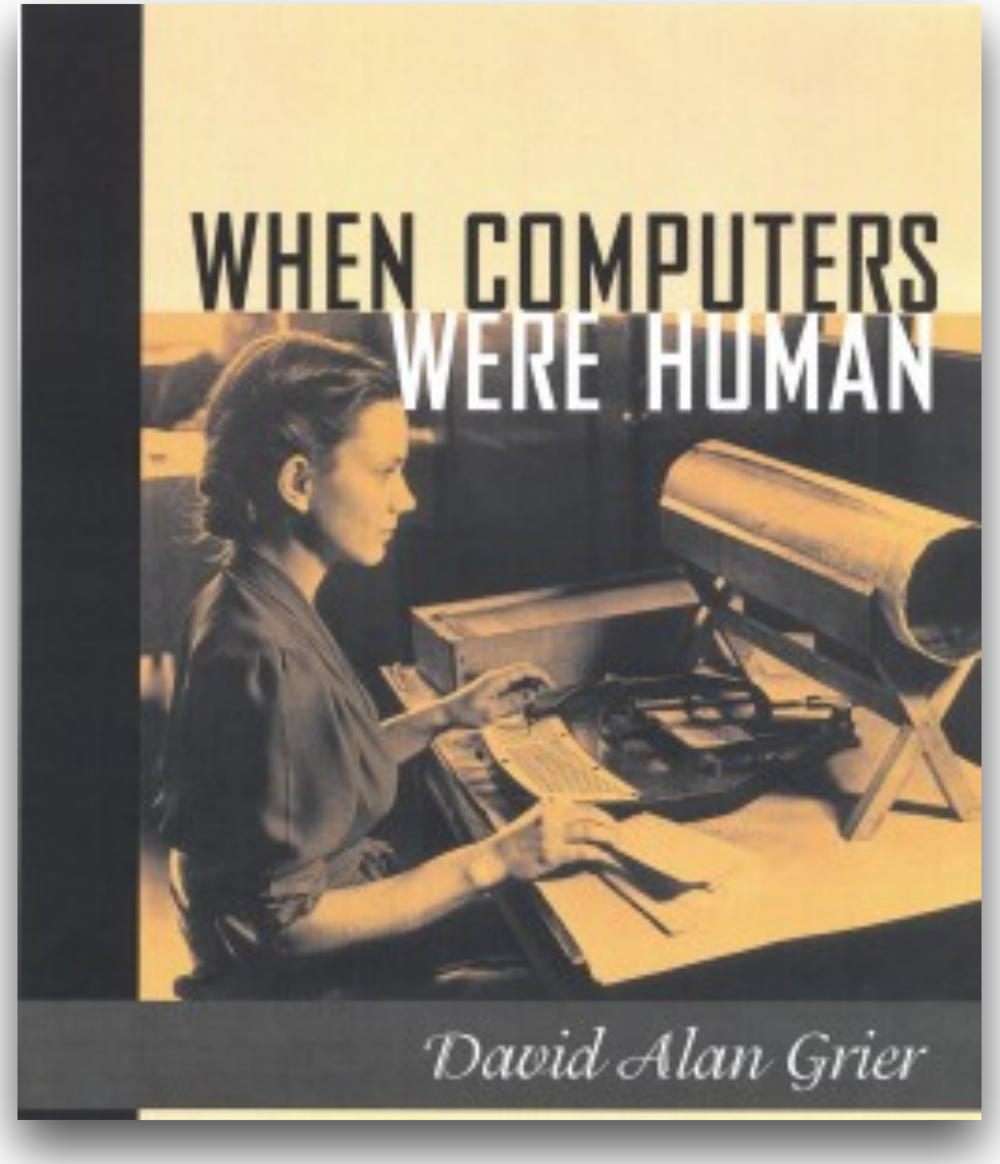
Decision problem

A formal model for *all* problems

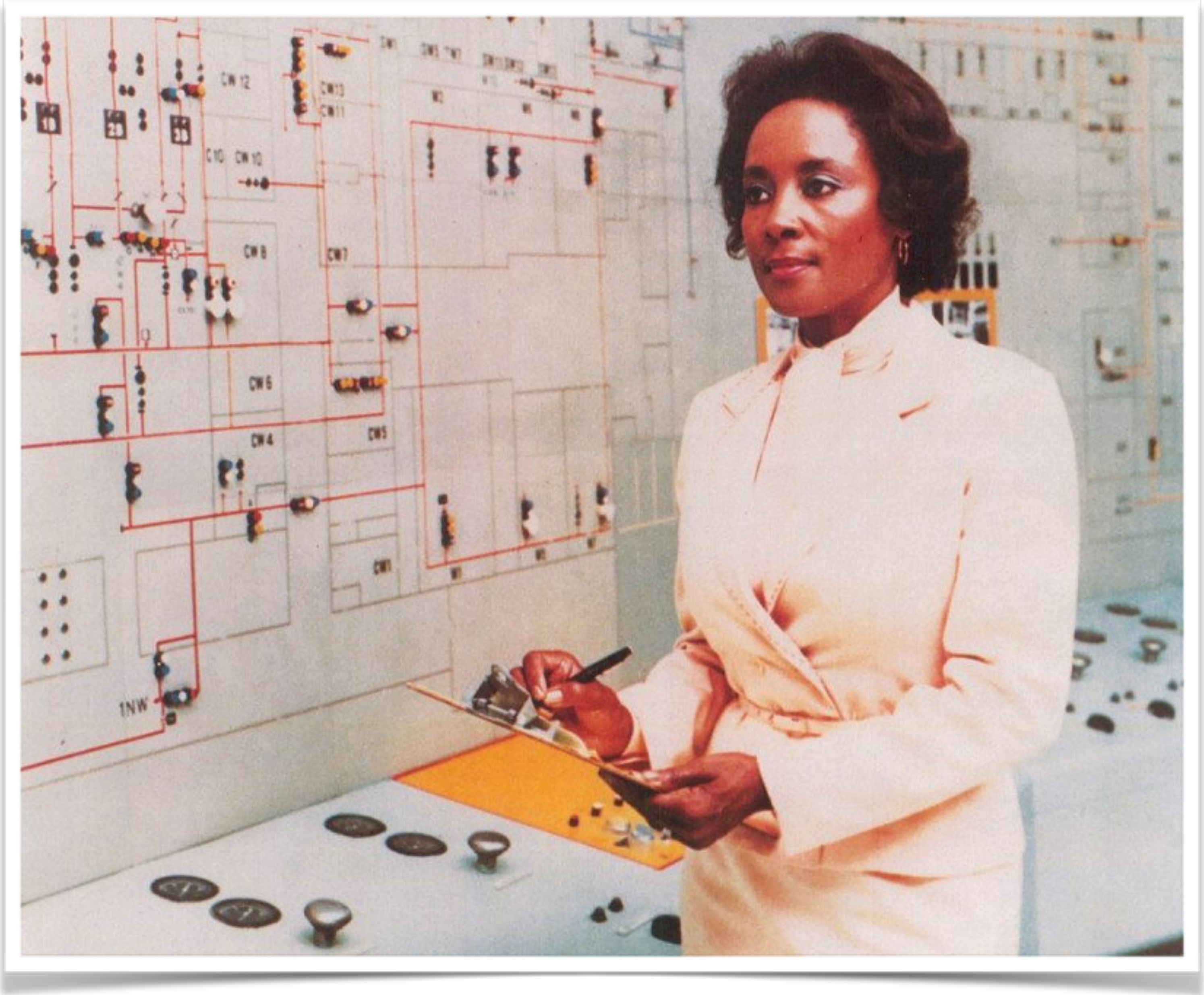
A yes-or-no question.

Ask yourself: can I encode *all* problems as yes-or-no questions?!

What do we mean by “computer”?



www.dagrier.net/wp-content/uploads/2013/04/grier.when-computers-were-human-259x300.jpg



https://d1o50x50snmhul.cloudfront.net/wp-content/uploads/2017/01/20120618/annie-easley-800x649.jpg

Annie Easley

What do we mean by “computer”?

Operating System (CP/M, DOS, Windows 8, Windows 10, MacOS, iOS 10, iOS 11,…)

Processor clock rate (1 MHz, 3.2MHz, 2.8 GHz, 8.79433 GHz, …)

Memory capacity (4KB, 64KB, 1MB, 8MB, 4GB, 512GB, …)

Power source (electricity, natural gas, solar, dilithium, …)

Construction materials (silicon, graphene, legos, …)

Programming language (Python, Java, Racket, …)

Architecture (single core, multicore, pipelined, out-of-order, branch predicting, GPU, VLIW, …)

Data representation (binary, trinary, ASCII, Unicode, …)

we need a
"computational model"

Today’s model: a **state machine**

Reads the **input** string one **symbol** at a time.
we will often use bitstrings for the input

Has a set of possible “configurations” (**states**).

Has rules for how to **transition** from one state to another.
based on current state and current input symbol

Accepts ("yes") or **rejects** ("no"), based on the input so far.

What is a state?

What are the "states" of this system?

Finite state machines

A **finite state machine (FSM)** is a state machine with:

- a predetermined, finite-sized set of **states**
  think of each state as a subtask
- a predetermined, finite-sized **alphabet** of valid input characters
  we use the capital greek letter “Sigma” to denote the alphabet: Σ
- a set of rules that describe **transitions** from each state for each character
  so that each state knows what to do for any possible input character
- a designated **initial state**
  the state that the machine is in when it starts
- a set of **accepting states**
  which determine the conditions under which the machine says “yes”

Common vocabulary for state machines

computer scientists say

Deterministic **F**inite **A**utomaton
(**DFA**)

deterministic: each state has one (and only one) transition for each possible input character.

finite: there are a finite number of states.

automaton: it operates under its own power.

engineers say

Finite **S**tate **M**achine
(**FSM**)

We'll use both "DFA" and "FSM", interchangeably.

The “automation” part of DFAs

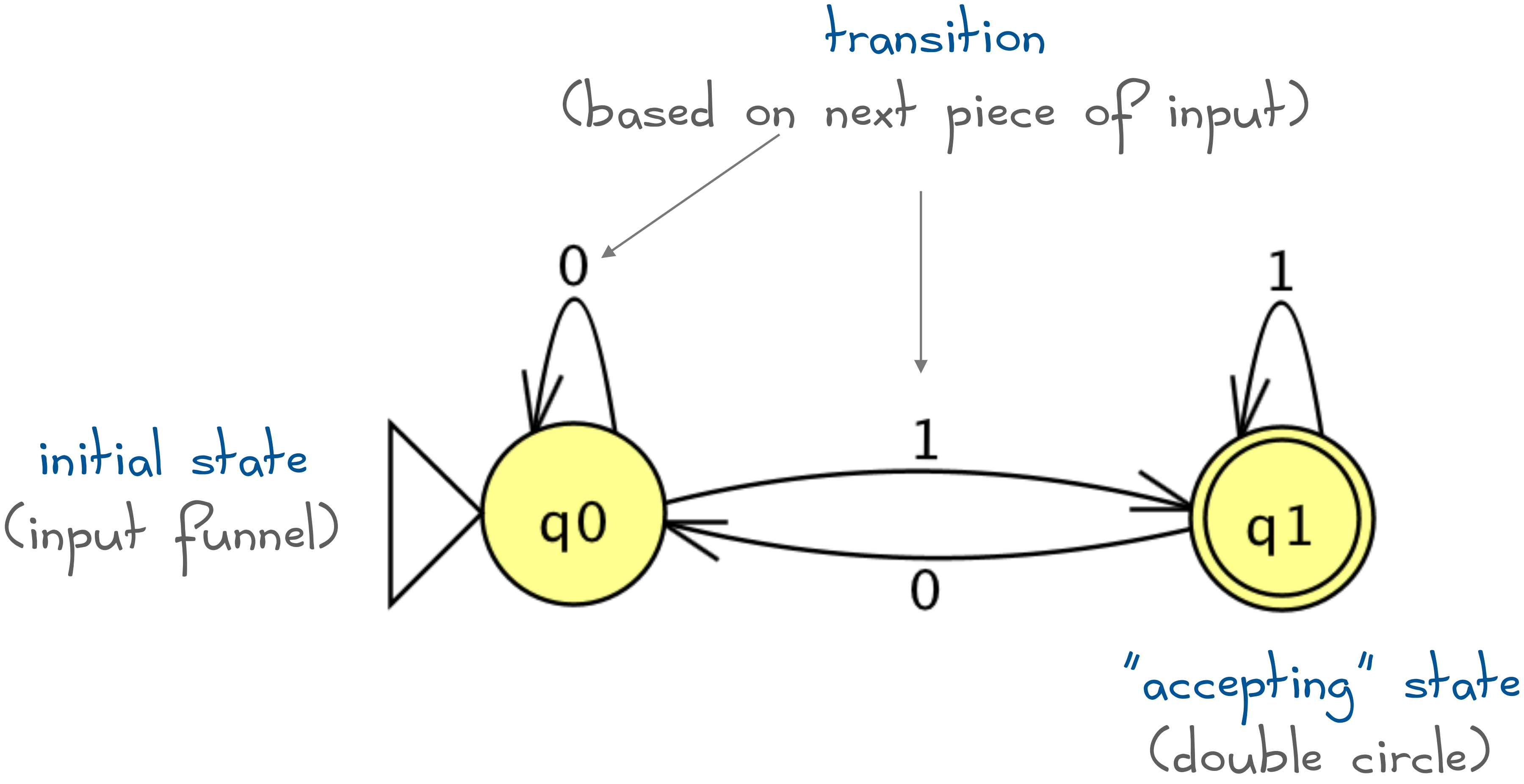
Reads the input string one symbol at a time.

Has a finite set of possible “configurations” (states).

Has rules for how to proceed from one state to another.
based on current state and current input

Accepts ("yes") or rejects ("no"), based on the input so far.

Stops when it has read all the input.



transition
(based on next piece of input)

initial state
(input funnel)

"accepting" state
(double circle)

DFAs describe a **language**—a set of strings

it’s all the inputs accepted by the DFA

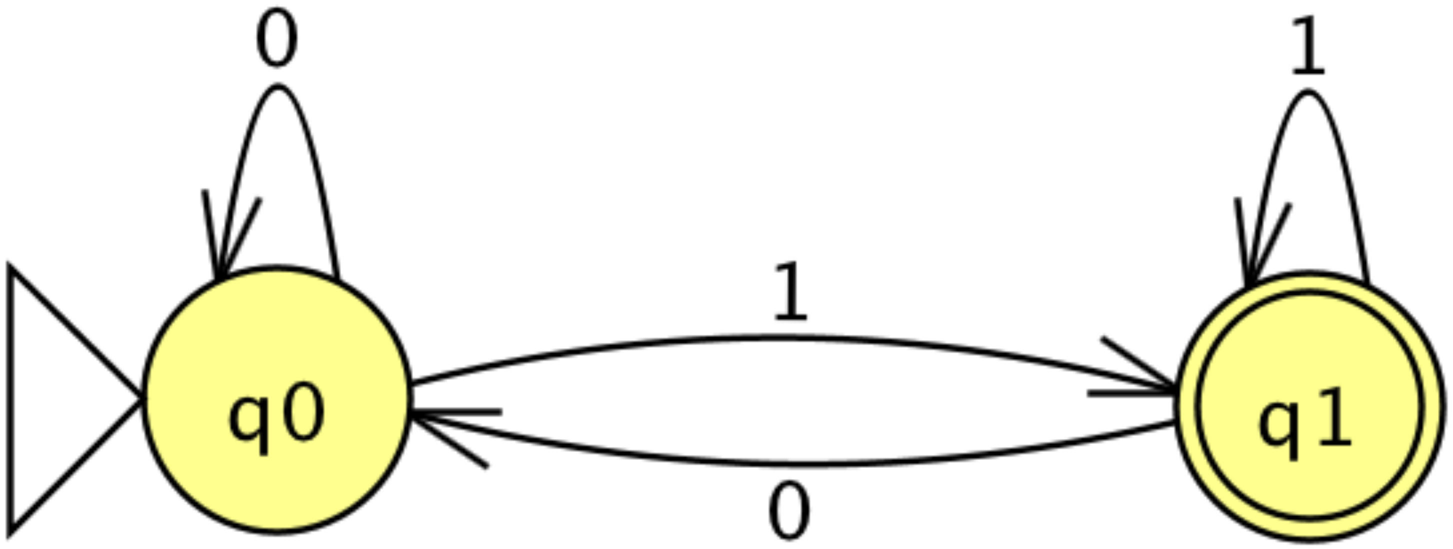
What language is described by this DFA?

L = {1, 01, 001, 011, …}

L = $\{w \mid w$ ends in a 1$\}$

“L is all strings w such that w ends in a 1”

(Bitstrings that encode) odd numbers



DFAs describe a **language**—a set of strings

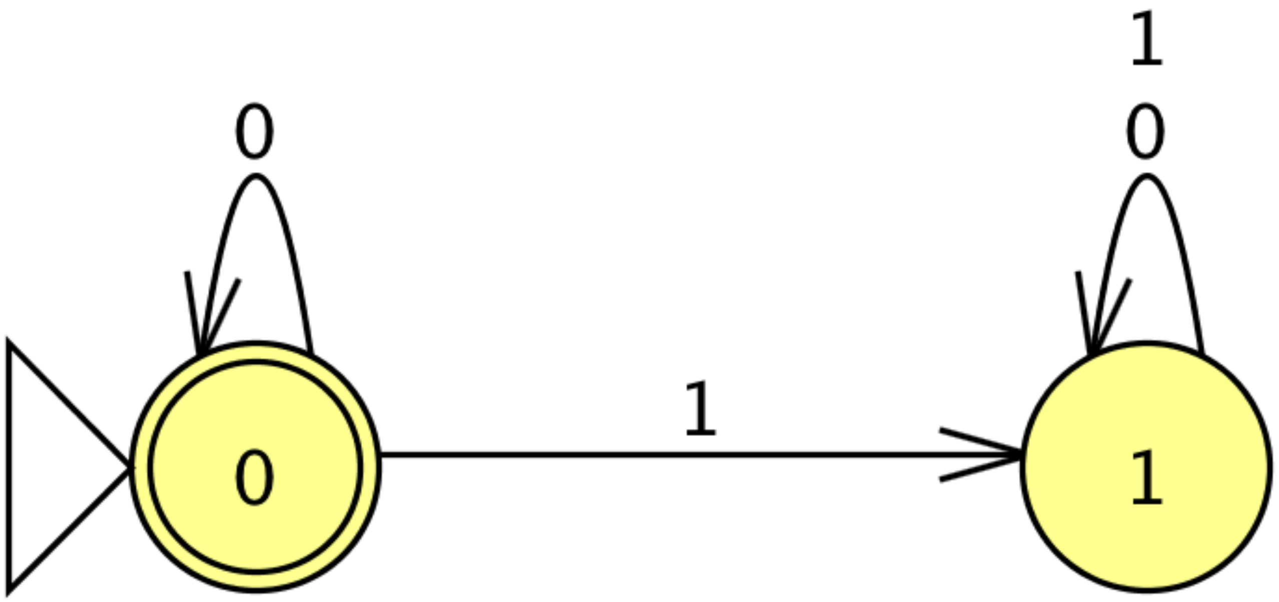
it’s all the inputs accepted by the DFA

What language is described by this DFA?

empty string

L = {λ, 0, 00, 000, …}

L = {w | w has no 1s}



Finite states = finite memory.

What are we “remembering” about the input so far) in each state?

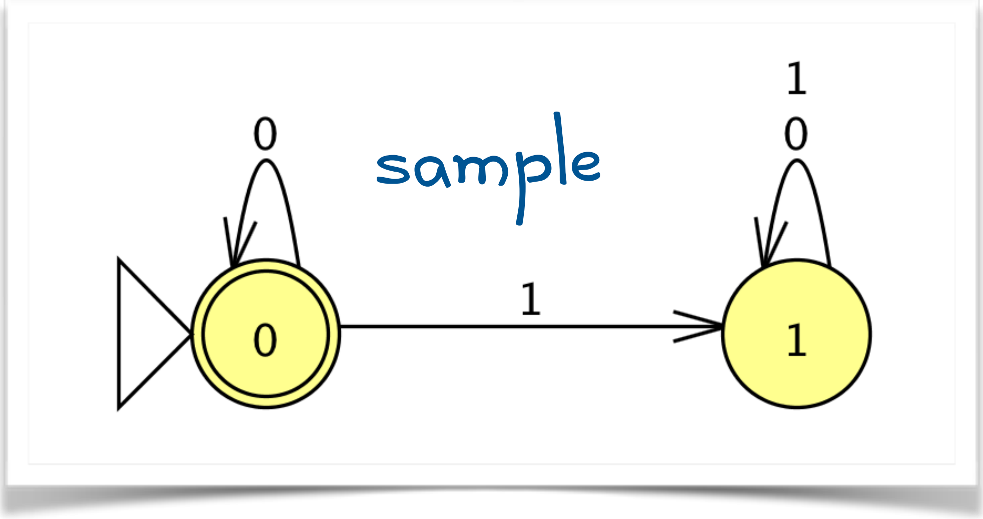
Draw these DFAs

It’s okay if you don’t finish the third one.

(1) L = {w | w contains at least two 1s}

(2) L = {w | the third bit in w is a 1}

(3) L = {w | the number of 0s in w is a multiple of 3}



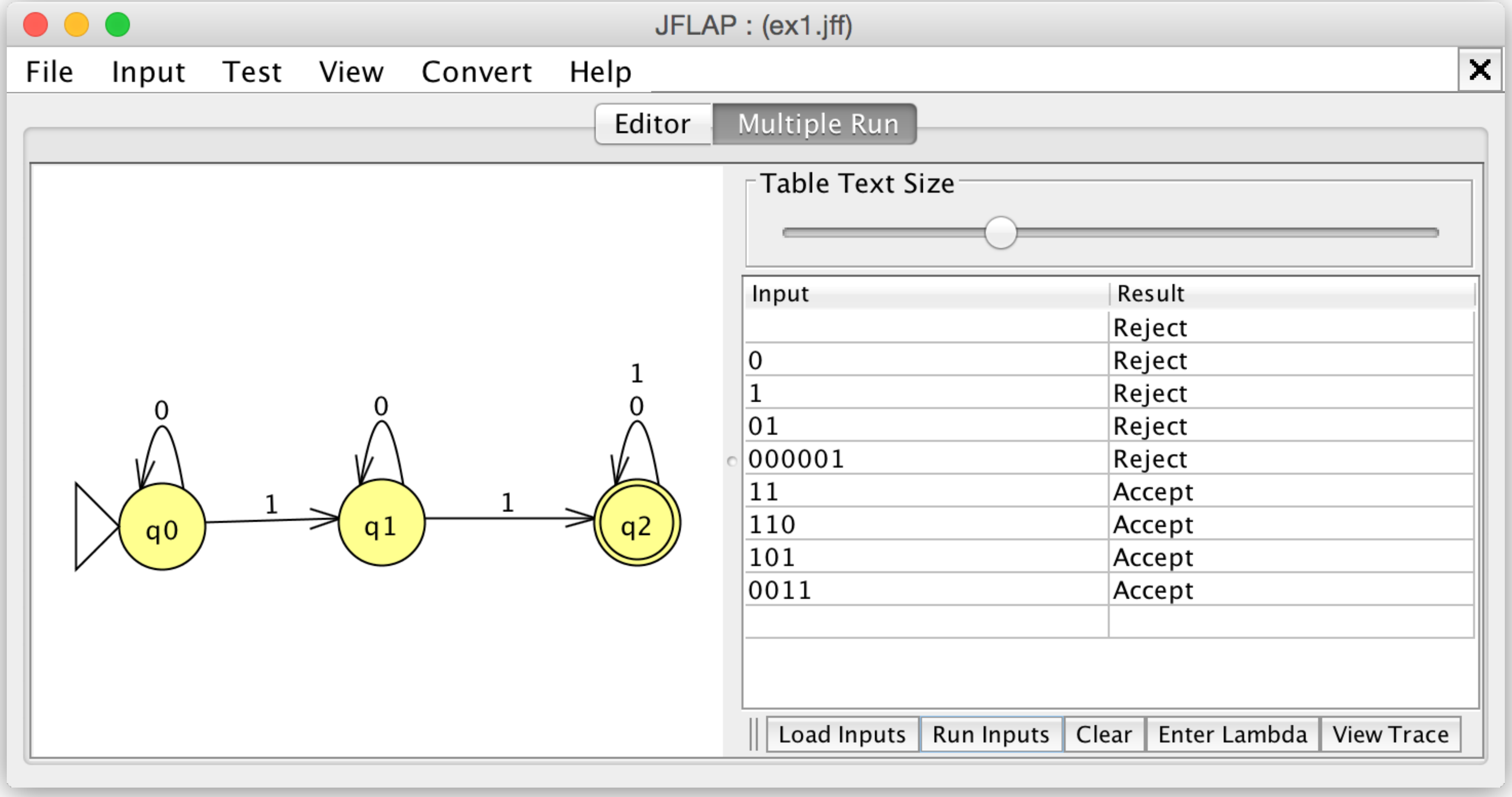
Write test cases first!!!!

At least three strings **accepted** by the DFA.

At least three strings **rejected** by the DFA.

JFLAP—a tool for making automata

(1) L = {w | w contains at least two 1s}



Input	Result
	Reject
0	Reject
1	Reject
01	Reject
000001	Reject
11	Accept
110	Accept
101	Accept
0011	Accept

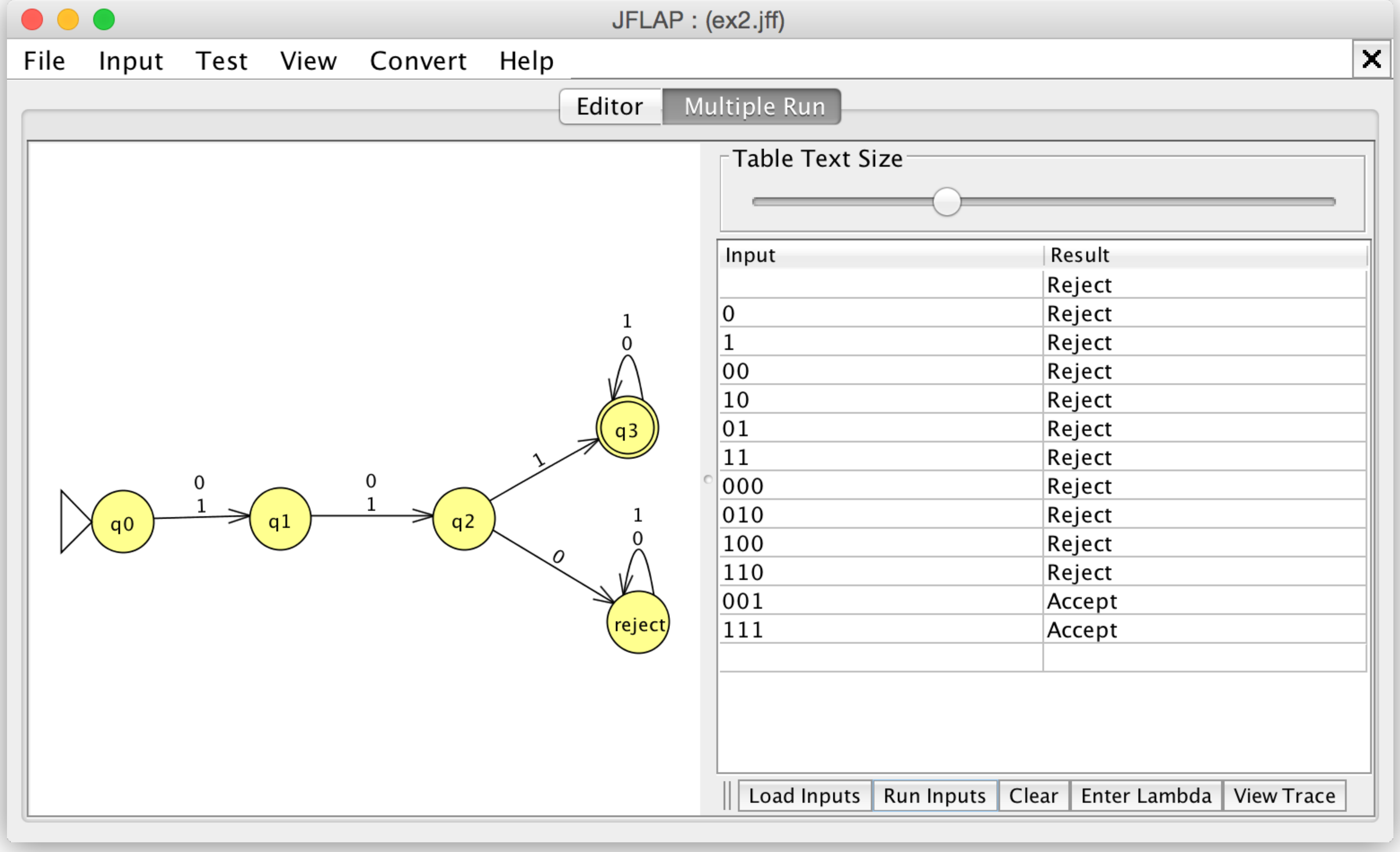
Write test cases first!!!!

At least three strings **accepted** by the DFA.

At least three strings **rejected** by the DFA.

JFLAP—a tool for making automata

(2) L = {w | the third bit in w is a 1}



Input	Result
	Reject
0	Reject
1	Reject
00	Reject
10	Reject
01	Reject
11	Reject
000	Reject
010	Reject
100	Reject
110	Reject
001	Accept
111	Accept

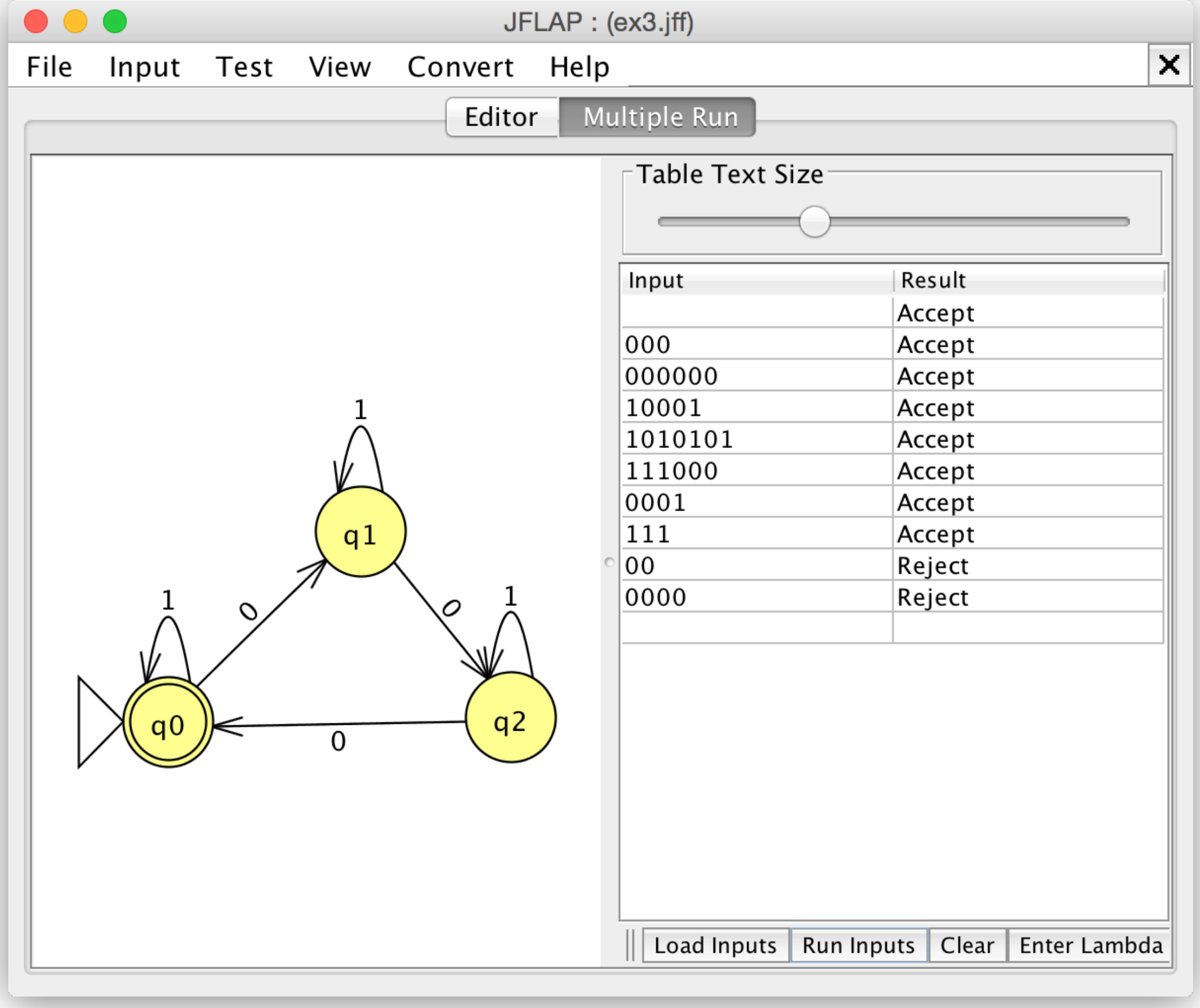
Write test cases first!!!!

At least three strings **accepted** by the DFA.

At least three strings **rejected** by the DFA.

JFLAP—a tool for making automata

(2) L = {w | the number of 0s in w is a multiple of 3}



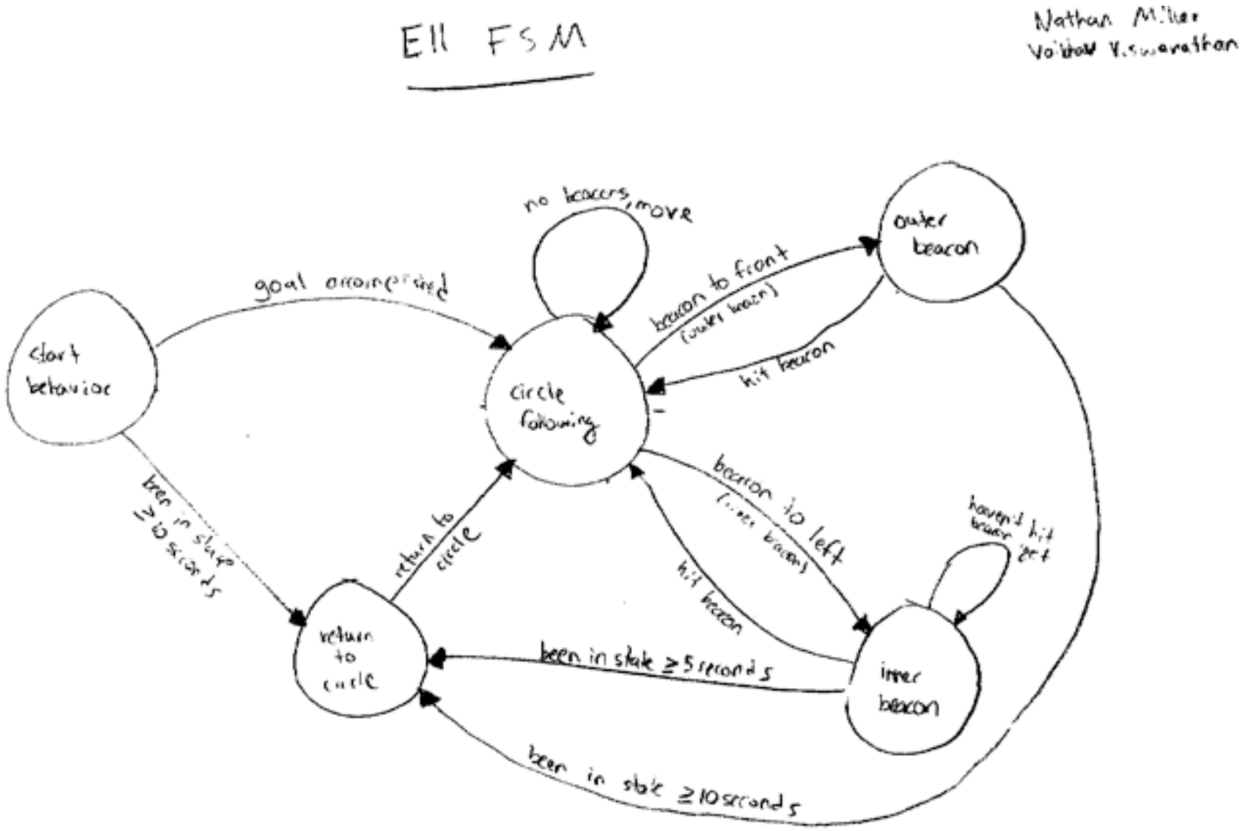
Input	Result
	Accept
000	Accept
000000	Accept
10001	Accept
1010101	Accept
111000	Accept
0001	Accept
111	Accept
00	Reject
0000	Reject

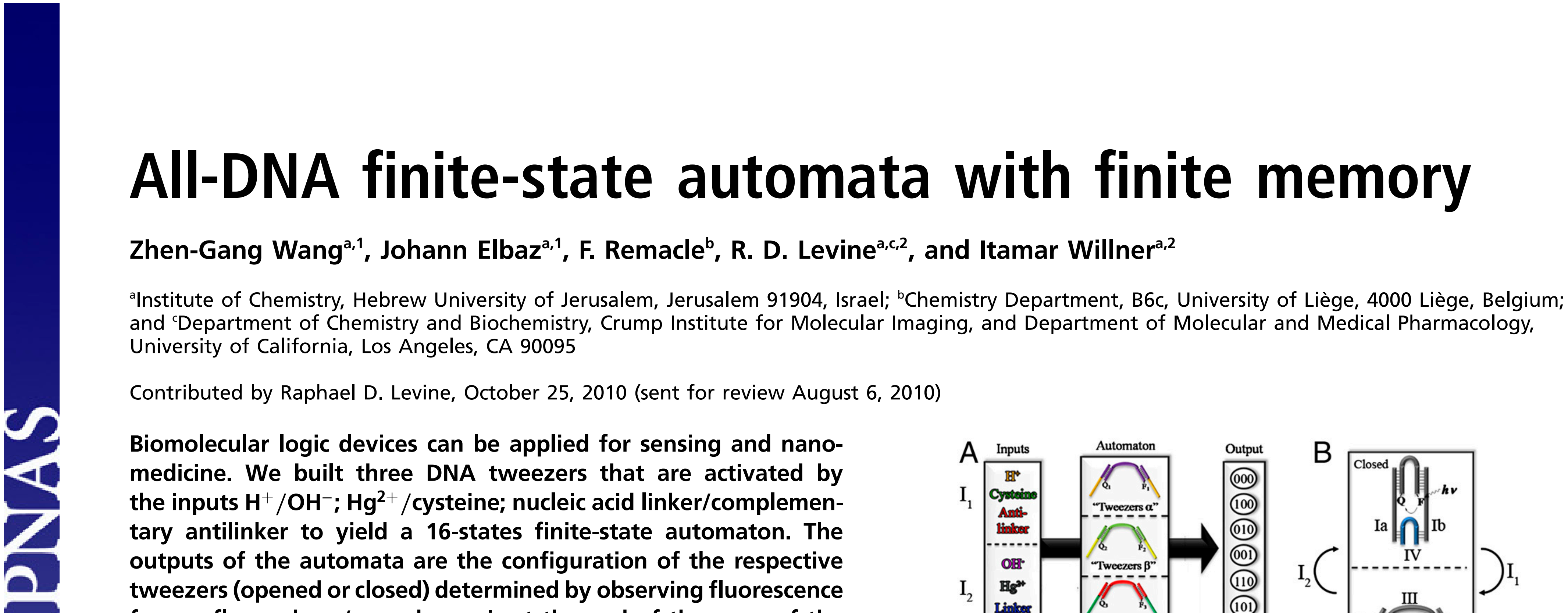
Write test cases first!!!!

At least three strings **accepted** by the DFA.

At least three strings **rejected** by the DFA.

DFAs are useful—they're everywhere!





All-DNA finite-state automata with finite memory

Zhen-Gang Wang[a,1], Johann Elbaz[a,1], F. Remacle[b], R. D. Levine[a,c,2], and Itamar Willner[a,2]

[a]Institute of Chemistry, Hebrew University of Jerusalem, Jerusalem 91904, Israel; [b]Chemistry Department, B6c, University of Liège, 4000 Liège, Belgium; and [c]Department of Chemistry and Biochemistry, Crump Institute for Molecular Imaging, and Department of Molecular and Medical Pharmacology, University of California, Los Angeles, CA 90095

Contributed by Raphael D. Levine, October 25, 2010 (sent for review August 6, 2010)

Biomolecular logic devices can be applied for sensing and nanomedicine. We built three DNA tweezers that are activated by the inputs H^+/OH^-; Hg^{2+}/cysteine; nucleic acid linker/complementary antilinker to yield a 16-states finite-state automaton. The outputs of the automata are the configuration of the respective tweezers (opened or closed) determined by observing fluorescence

First assignment

Available online later tonight—watch for Piazza message

Due next Tuesday at 11:59pm

- You can use 1 (of 3) “Euros" to turn in 24 hours later, no need to tell us
- I drop your lowest assignment score

Use JFLAP to make DFAs

Remember: pair-programming, office hours, tutoring hours

Test and turn in online—watch for Piazza message