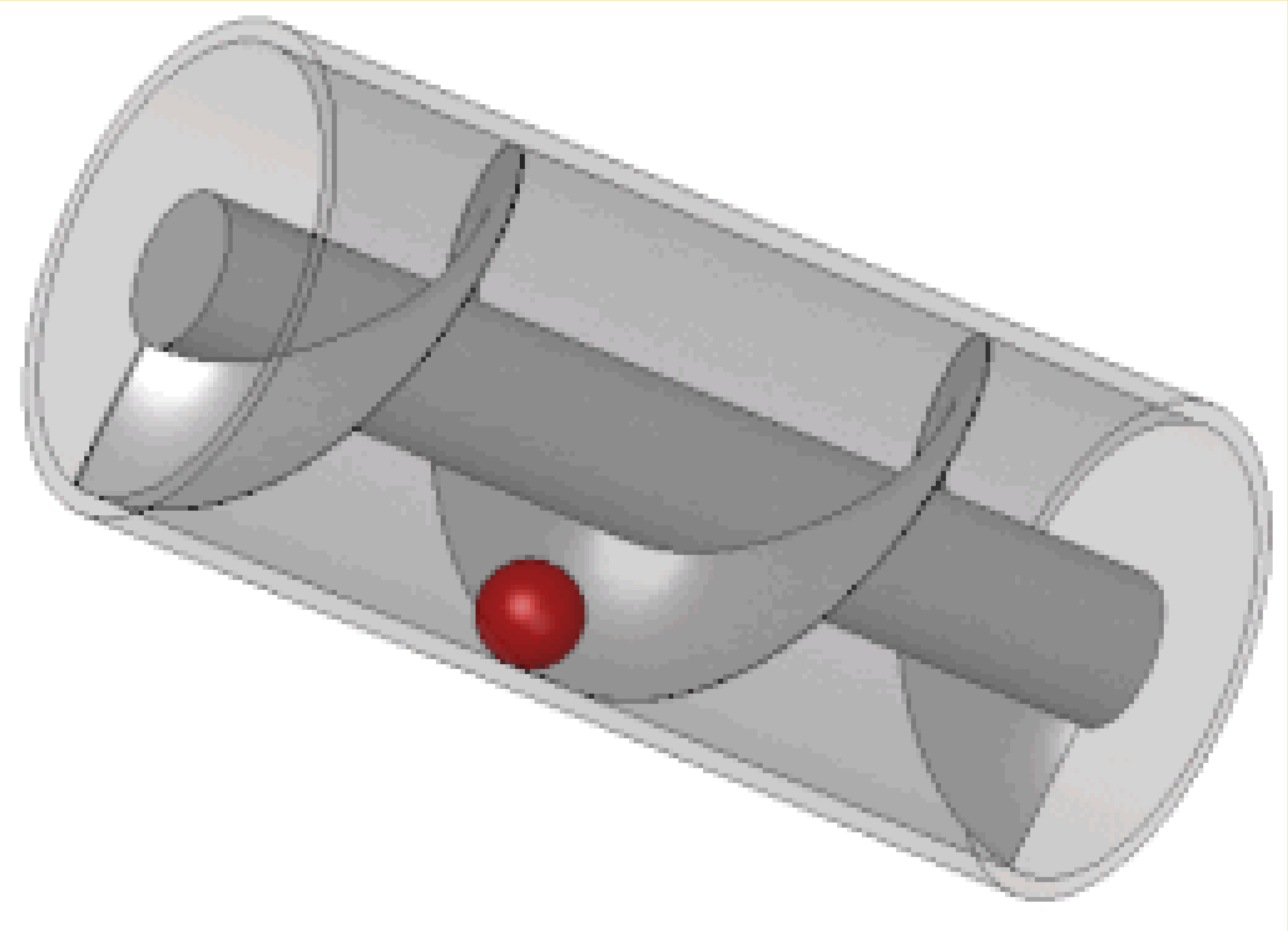
Midterm: Mar 6, 12:30pm
Same classroom
PS6 due Mar 21, 10pm



https://en.wikipedia.org/wiki/Archimedes%27_screw

Class 14: Limitations of Regular Expression

University of Virginia
cs3120: DMT2
Wei-Kai Lin

Definition 6.7 (Matching a regular expression)

Let e be a regular expression over the alphabet Σ. The function $\Phi_e : \Sigma^* \rightarrow \{0,1\}$ is defined as follows:

1. If $e = \sigma$ then $\Phi_e(x) = 1$ iff $x = \sigma$.
2. If $e = (e'|e'')$ then $\Phi_e(x) = \Phi_{e'}(x) \vee \Phi_{e''}(x)$ where $\vee$ is the OR operator.
3. If $e = (e')(e'')$ then $\Phi_e(x) = 1$ iff there is some $x', x'' \in \Sigma^*$ such that x is the concatenation of x' and x'' and $\Phi_{e'}(x') = \Phi_{e''}(x'') = 1$.
4. If $e = (e')*$ then $\Phi_e(x) = 1$ iff there is some $k \in \mathbb{N}$ and some $x_0, \ldots, x_{k-1} \in \Sigma^*$ such that x is the concatenation $x_0 \cdots x_{k-1}$ and $\Phi_{e'}(x_i) = 1$ for every $i \in [k]$.
5. Finally, for the edge cases $\Phi_\emptyset$ is the constant zero function, and $\Phi_{\texttt{""}}$ is the function that only outputs 1 on the empty string `""`.

We say that a regular expression e over Σ *matches* a string $x \in \Sigma^*$ if $\Phi_e(x) = 1$.

Theorem:
Reg-Fun $=$ DFA-Comp

Theorem 6.17 (DFA and regular expression equivalency)

Let $F : \{0,1\}^* \to \{0,1\}$. *Then* F *is regular if and only if there exists a DFA* $(T, \mathcal{S})$ *that computes* F.

Recap: Regular Expressions $\equiv$ Finite Automata

Definitions:
Reg-Fun: the set of all regular functions.
DFA-Comp: the set $\{f \mid f$ is computed by some DFA $M\}$

Theorem:
Reg-Fun $=$ DFA-Comp

Recap: Regular Expressions $\equiv$ Finite Automata

Interpretation:

For any $L \subseteq \{0,1\}^*$,
we have a regular expression e such that matches L iff
we have a DFA M such that matches L.

- Negate any regular expression
- OR any DFA

Limits of finite state computation

There is at least one non-regular function

- Proof: e string of 8 spec symb.
- Reg-Fun is a countable set (why?) DFA-Comp
- Set of all functions (All-Fun) is uncountable (why?)

  $f : \{0,1\}^* \to \{0,1\}$

- If All-Fun ⊆ Reg-Fun, then All-Fun would have been countable

Any "natural" languages that is not regular?

- By Reg-Fun = DFA-Comp,
  Any regular function must be computable by a DFA
- Any DFA has constant "memory", ie, num of states

- Find a function needs more than const mem

Example $A = \{0^k 1^k : k \in N\}$

$\underbrace{(((((}_{k}\quad \underbrace{)))))}_{k}$

Theorem: A is not regular

Proof: Assume for contrad. $\exists$ DFA M accepts computes A

$x = \underbrace{111\ldots\ldots 1}_{n} \quad n > m$

$y = 0^n$

$xy \in A$

$\Rightarrow M$ acc: $\underbrace{1^j 1^{2i} 1^{n-j-i}}_{j+2i+n-j-i \,=\, n+i \,>\, n} 0^n$

m states

$\Rightarrow j+i<n$

$i \geq 1$

contra $\square$

Example A = $\{\mathbf{0}^{k}\mathbf{1}^{k}\text{: } k \in \mathbf{N}\}$

Theorem: A is not regular

Proof:

Give it 00 … 0 … as inputs till a state q is repeated.

Let $x = 0^i, y = 0^j$ such that $j > 0$ and
both x and $x \mid\mid y$ on q.

Consider $0^i 1^i$ and $0^{i+j} 1^i$.

Either both will be accepted,

Or both will be rejected.

Pumping Lemma

Let e be a regular expression over the alphabet of bits, $\{0,1\}$.

There is some number n_0 such that
for every $w \in \{0,1\}^*$ with $|w| > n_0$ and $\Phi_e(w) = 1$,
we can write $w = xyz$ for strings $x, y, z \in \{0,1\}^*$
satisfying the following conditions:

- $|y| \geq 1$
- $|xy| \leq n_0$
- $\Phi_e(x\, y^k\, z) = 1$ for every natural number k

Proof

$\forall e \; \exists$ DFA M s.t. $\Phi_e(x) = M(x)$.

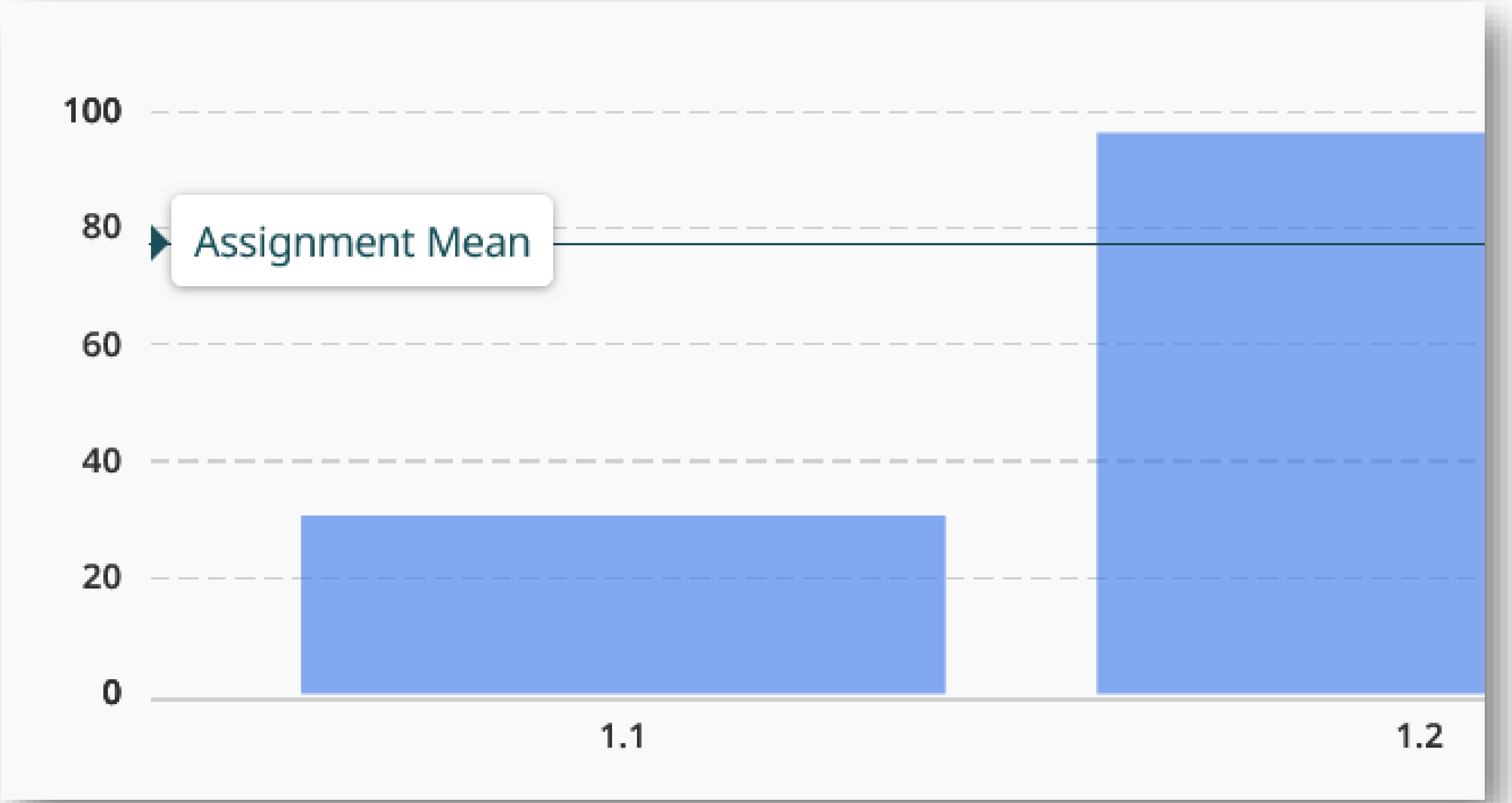
Look at num of states of M

If M acc w st. $|w| > n_0$

Then w visits the same state q twice.

$w = xyz$, $|y| \geq 1$

PRR6: Q1.1



What is asked?

Q1.1

2 Points

Notation: $n_0 = p$ and $w = s$

For every DFA M, there exists a length p and a string $s = xyz$ of length at least p such that satisfy the following:

For all

- $M(s)$ accepts
- for each $i \geq 0$, $M(xy^iz)$ accepts
- $|y| > 0$
- $|xy| \leq p$

$w = \underbrace{1}_{x}\,\underbrace{1011000}_{y}\,\underbrace{\;}_{z}$

$y = ""$ $|y| = 0$

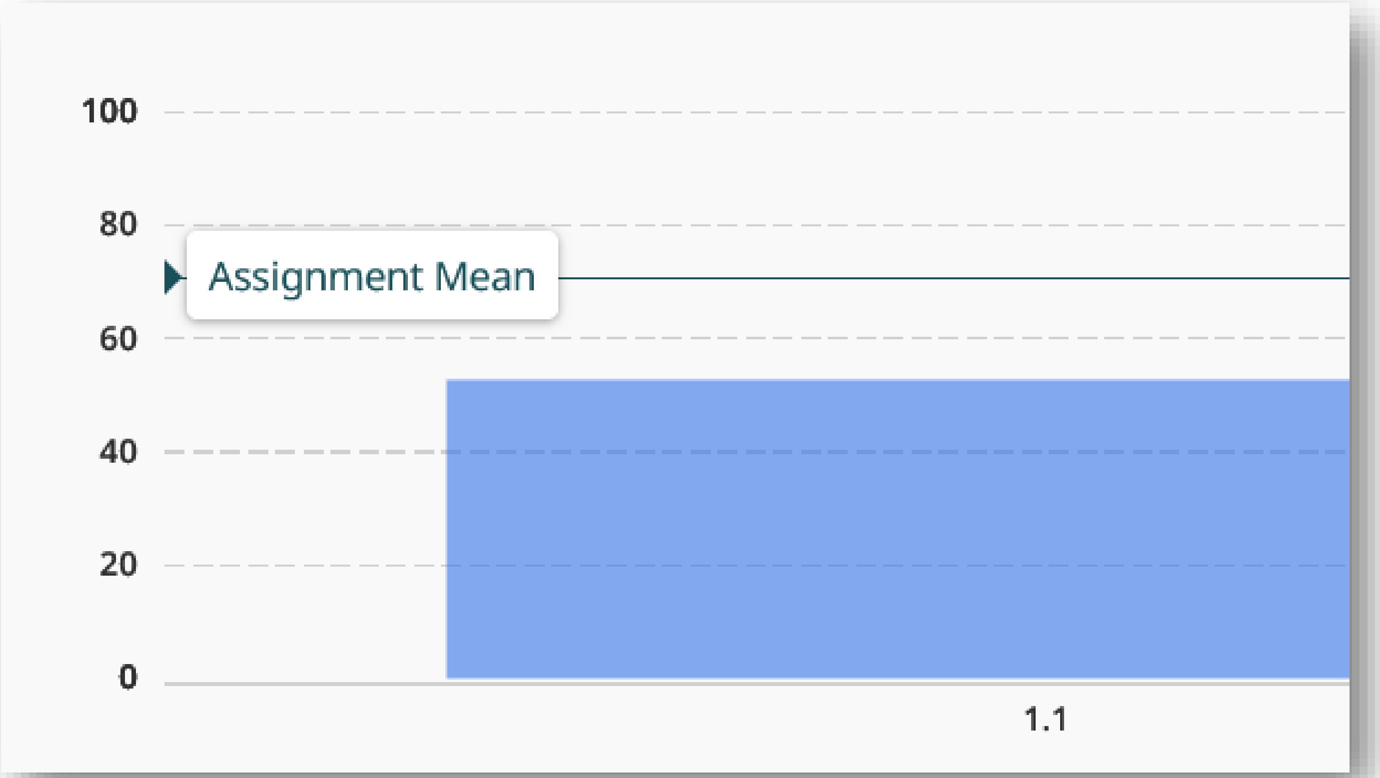
○ Yes

◉ No

Consider M that accepts only the string "11011000".

What's n_0 and w? NO

PRR7: Q1.1



What is asked?

Q1.1
5 Points

Let S be the set of binary strings such that S consists of strings that are exactly repeating 1s for $2n$ times for some natural number n. Namely, {empty_string, 11, 1111, 111111, 11111111, ...} are the shortest strings in S for $n = 0, 1, 2, 3, 4, \ldots$. Equivalently and formally,

$$S = \{x \in \{0,1\}^* : x = 1^{2n} \text{ for some natural number } n\}$$

Is S a regular language? Equivalently, is there a regular expression that matches strings in S?

- (●) Yes
- ( ) No

Regular expression: $(11)^*$
(Exercise: draw a DFA)

Intuition: this is just "counting to 2"
(which is finite)

Q1.2
5 Points

Let S be the set of binary strings such that S consists of strings that are exactly repeating 1s for 2^n times for some natural number n. Namely, 1, 11, 1111, 11111111, 1111111111111111, ... are the shortest strings in S for $n = 0, 1, 2, 3, 4, \ldots$. Equivalently and formally,

$$S = \{x \in \{0,1\}^* : x = 1^{2^n} \text{ for some natural number } n\}$$

Is S a regular language? Equivalently, is there a regular expression that matches strings in S?

NO.
How to prove it?

Not Regular: $S = \{\, x \in \{0,1\}^* \;:\; x = 1^{2^n}$ for some natural number $n\}$

Proof: Assume for contra, $\exists e$ s.t. matches S

$\exists n_0 \;\forall w$ s.t. .... $w = x\,y\,z$

$w = 1^{2^n} = \underbrace{1 \cdots\cdots 1}_{2^n}$ s.t. smallest $2^n > n_0$

find x, y, z ? $x\,y = 1^{n_0 - a}$

choose $k = 2$ $x y^2 z = |1^{2^n} \cancel{+} y| \cancel{\leq 2}$

$\leq 2^{n+1}$

Let e be a regular expression over the alphabet of bits, $\{0,1\}$.
There is some number n_0 such that for every $w \in \{0,1\}^*$ with $|w| > n_0$ and $\Phi_e(w) = 1$,
we can write $w = xyz$ for strings $x, y, z \in \{0,1\}^*$ satisfying the following conditions:

- $|y| \geq 1$
- $|xy| \leq n_0$
- $\Phi_e(x\,y^k\,z) = 1$ for every natural number k

$xy^kz \in S$?

Midterm: this Thursday, 12:30pm
Rice 130

XOR is not Universal

Problem 2 *XOR is not universal.*

Prove that for every n-bit input circuit C that contains only XOR gates, as well as gates that compute the constant functions ZERO and ONE, C is affine or linear modulo two, in the sense that there exists some $a \in \{0,1\}^n$ and $b \in \{0,1\}$ such that for every $x \in \{0,1\}^n$, $C(x) = \sum_{i=0}^{n-1} a_i x_i + b \mod 2$.

Conclude that the set $\{\text{XOR}, \text{ZERO}, \text{ONE}\}$ is not universal.

Two sub-questions:

1. Any XOR-circuit computes a linear function (mod 2)
2. Any linear function cannot compute NAND

Any XOR-circuit computes a linear function

Induction on circuit depth d.

Predicate, P(d):

Any XOR-circuit of depth d computes a linear function

Base case: P(0) holds by no XOR, only input

Inductive case: P(d) holds, want P(d+1)

1. Let K(x) be a (d+1)-depth XOR circuit
2. $K(x) = XOR(K_1(x), K_2(x))$
3. K_1, K_2 depth d
4. $f_1(x), f_2(x)$ linear
5. $K(x) = XOR(f_1(x), f_2(x))$ linear

$$\begin{array}{ccc} & XOR & \\ \nearrow & & \nwarrow \\ K_1(x) & & K_2(x) \end{array}$$

Any linear function cannot compute NAND

Let $f(x_1, x_2) = a_1\, x_1 + a_2 x_2 \mod 2$

Want NAND(x_1, x_2) = $f(x_1, x_2)$ for all $x_1, x_2 \in \{0,1\}$

Charge

Regular Expressions

Limitations

Pumping Lemma

Midterm this Thursday, 12:30pm
PS6: due Mar 21, 10pm