Photo: Movie Poster

HW 3 due this Friday, Feb 13 (10:00pm)
HW 4 coming soon

Quiz 4 due next Monday, Feb 16, 10pm

Class 9:
NFA $\Rightarrow$ DFA

University of Virginia
CS3120: DMT2
https://weikailin.github.io/cs3120-toc
Wei-Kai Lin

Plan

Midterm 0

Grades, questions

Reg-Fun $\subseteq$ DFA-Comp

Non-deterministic FA

NFA $\subseteq$ DFA-Comp

Circuits

Today: [Sipser] Section 1.2

- Formal definition of a nondeterministic finite automaton
- Equivalence of NFAs and DFAs

Midterm 0

~~*Introduction to the Theory of Computation*~~
~~Section 1.2. Michael Sipser.~~

True, False, or Unknown?

(d) If ~~A is~~ $\mathbb{N}$, which is countably infinite, then the set $S = \{a \subseteq A \mid |a| \in \mathbb{N}\}$ is countable.
Note: S consists of all finite subsets of $\mathbb{N}$, which will be used frequently in CS 3120, e.g., representing graphs.

A is countable

$\overset{\mathbb{N}}{\overset{\shortparallel}{A}}$

Is S the same as $pow(A)$?

$A \in Pow(A)$

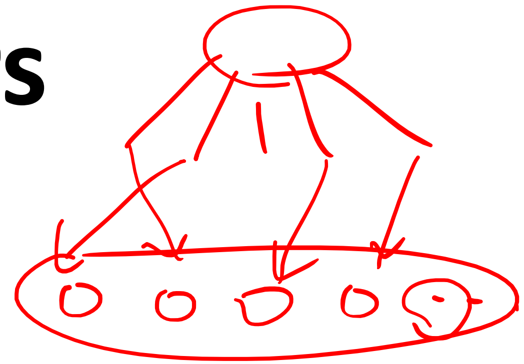
$|A| \notin \mathbb{N}$

EVEN

Consider the cardinality of S and
$T = \{(a_0, a_1, \ldots, a_n) \mid a_i \in \mathbb{N}, n \in \mathbb{N}\}$, where $|T| = |\mathbb{N}|$

$|S| \leq |T| \leq |\mathbb{N}|$

Proof: Cantorvanian Fingers



Fingers and Holes are countably infinite.

Q: How many mappings Fingers → Holes? $|S|$?

$$S = \{f\colon \mathbb{N} \to \mathbb{N} \mid \forall x \neq x', f(x) \neq f(x')\}$$

The Cantorvanian creatures from the planet Cantorvania have only one hand, but it has a countably infinite number of fingers. Hence, a Cantorvanian glove has a countably infinite number of finger holes. Suppose that each finger is unique (that is, each finger is assigned to a unique label) and that each finger hole is unique similarly. To wear a Cantorvanian glove on a Cantorvanian hand, we require that each finger must be put into one finger hole, and two distinct fingers must be put into distinct finger holes. It is okay to put any finger into any finger hole as long as the requirement is satisfied.

Show that there is an uncountable number of ways for a Cantorvanian to wear a Cantorvanian glove. You must use the definitions below, and you may use the theorem: A set S is countably infinite if and only if there is a bijection between S and $\mathbb{N}$.

Proof: Cantorvanian Fingers

Fingers and Holes are countably infinite.

Q: How many mappings Fingers → Holes? $|S|$?

$$S = \{\mathrm{f}: \mathbb{N} \to \mathbb{N} \mid \forall \mathrm{x} \neq \mathrm{x}', \mathrm{f}(\mathrm{x}) \neq \mathrm{f}(\mathrm{x}')\}$$

In class

Scores and Grades

- We know it is hard to formalize a proof in 20 min.
  - We tend to make shorter questions in the future

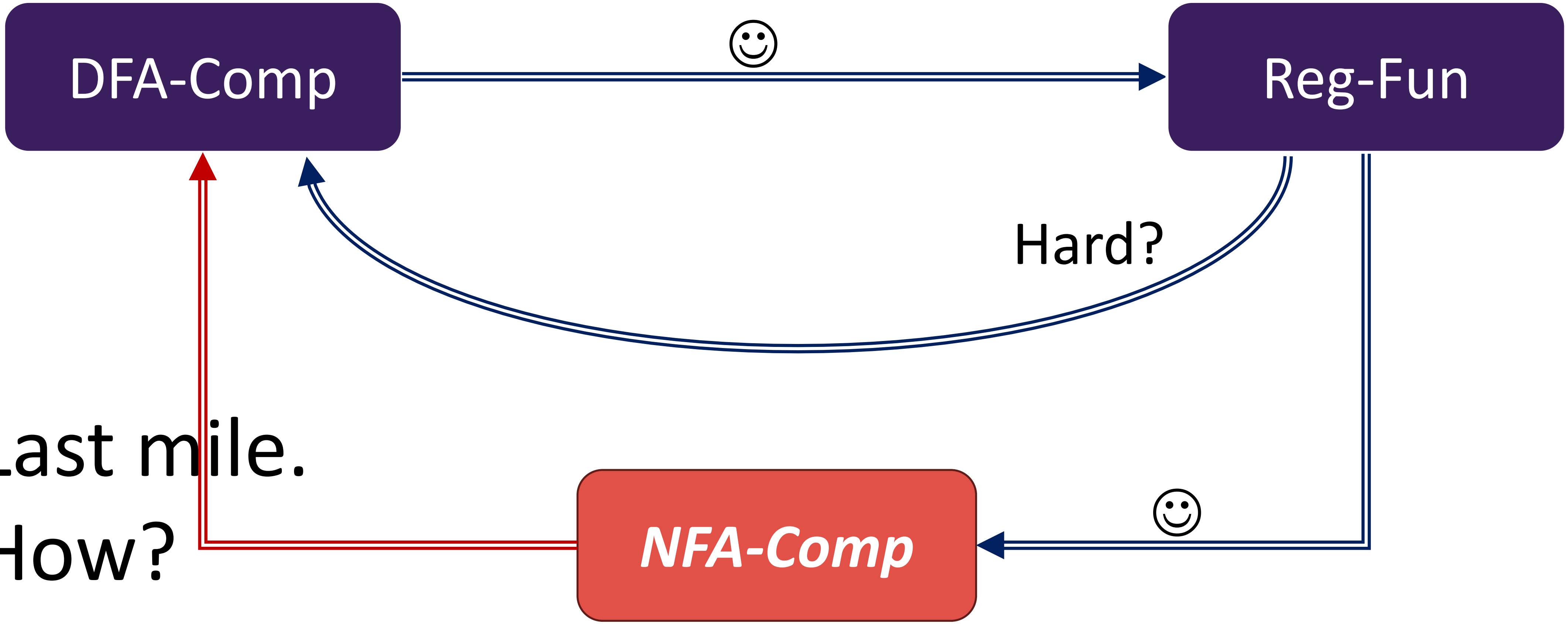
- An estimation:
  - Dropping lowest 1 out of 4
    ➔ median increase ~10% (std dev: 22%)
  - (HW + Quiz) + (Exam * 70%)
    32 + 68 * 70% = 83

Questions?

Reg-Fun $\subseteq$ DFA-Comp

Introduction to the Theory of Computation
Section 1.2. Michael Sipser.

Recall: High-Level Proof Plan



Last mile.

How?

Recall: Non-deterministic FA

A **Nondeterministic *Finite Automaton*** over alphabet {0,1} is a tuple (C, T, S) where:

1. C --- the number of *states*
2. $T: [C] \times \{0,1\} \rightarrow \boldsymbol{pow}([\boldsymbol{C}])$ a transition function
3. $S \subseteq [C]$ --- the set of accept states

Execution:

The string $x = b_0 b_1 \ldots b_n$ is matched by the NFA $M = (C, T, S)$ iff there are states $s_0, s_1, s_2, \ldots, s_n \in Q$ such that

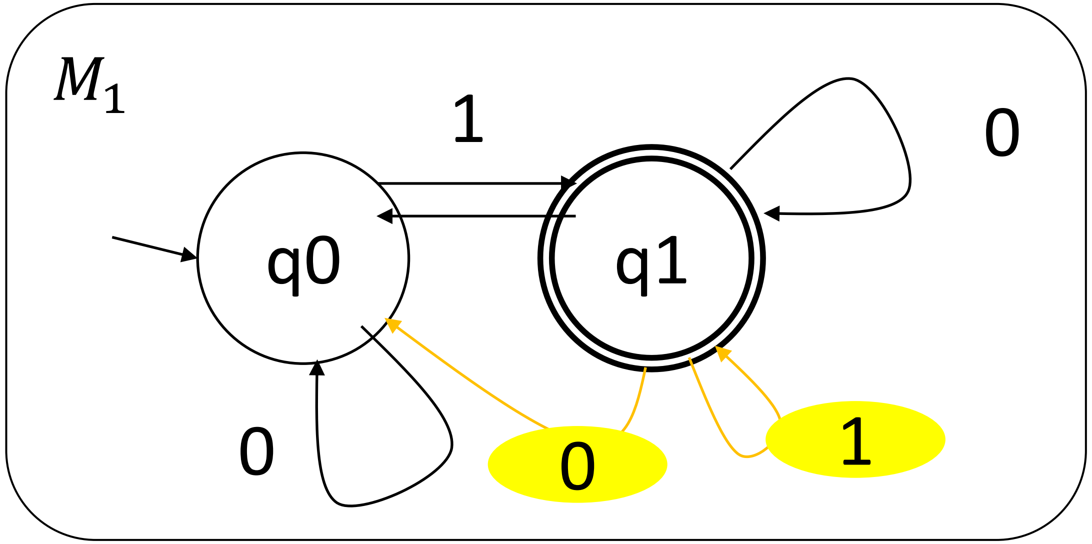
$\boldsymbol{s_{i+1}} \in \boldsymbol{T}(\boldsymbol{s_i}, \boldsymbol{b_i})$ for all $i = 0, \ldots, n-1$ and

$s_0 = 0$ and

$s_n \in S$.

Recall: Non-deterministic (Big Idea)

Suppose M_1 is equivalent to e_1



Allow transition to **multiple states** (clearly, not DFA)
Accept if exist a **path to accept**

NFA-Comp $\subseteq$ DFA-Comp

Introduction to the Theory of Computation
Section 1.2. Michael Sipser.

Power of NFA/DFA

$M_\epsilon \Rightarrow$ Reg Exp $\Rightarrow$ NFA

1. Is there any function a **DFA** can compute that cannot be computed by an **NFA**? No $\Rightarrow$ $M \in NFA$

DFA $T: [C] \times \{0,1\} \to [C]$

$T'(s, \alpha) := \{T(s,\alpha)\}$

2. Is there any function an **NFA** can compute that cannot be recognized by a **DFA**?

Power of NFA/DFA

1. Is there any function a **DFA** can compute that cannot be computed by an **NFA**?

DFA-Comp $\subseteq$ NFA-Comp

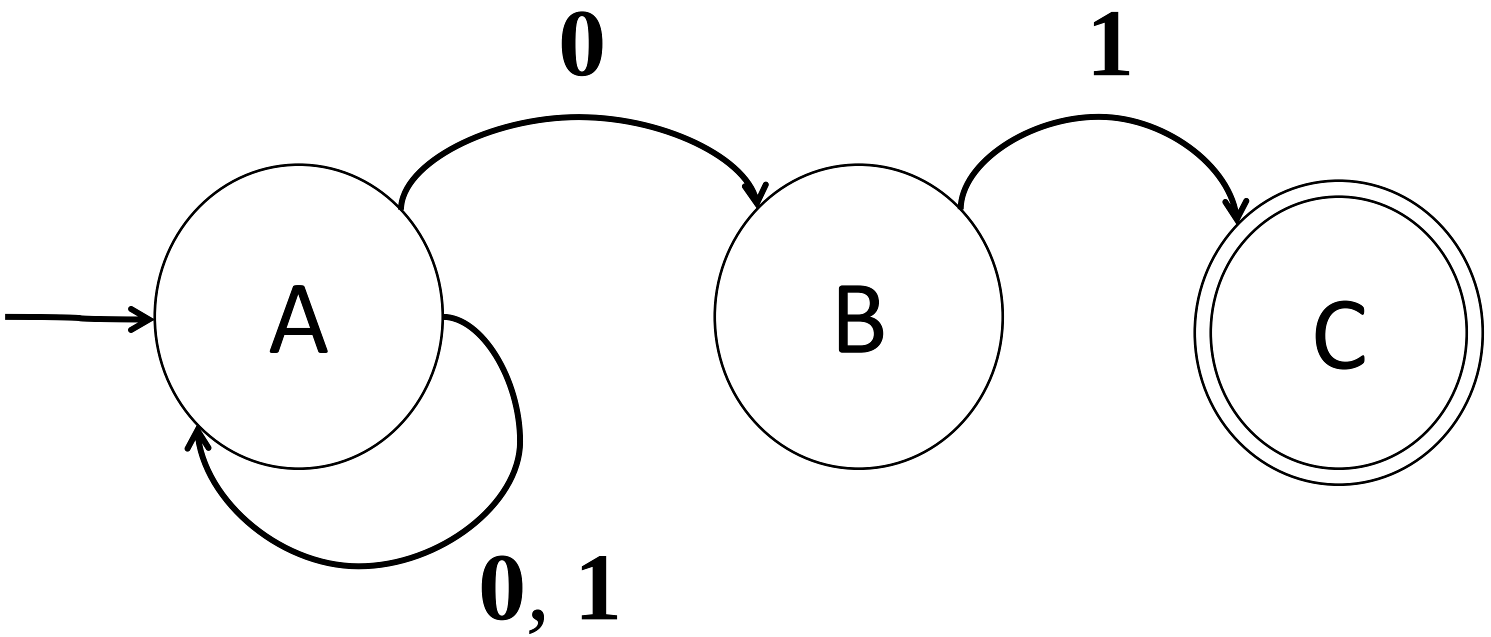
No: NFAs are at least as powerful as DFAs.

$$\text{NFA–Comp} \supseteq \text{DFA–Comp}$$

2. Is there any function an **NFA** can compute that cannot be recognized by a **DFA**?

$$\text{NFA–Comp} \subseteq \text{DFA–Comp}$$

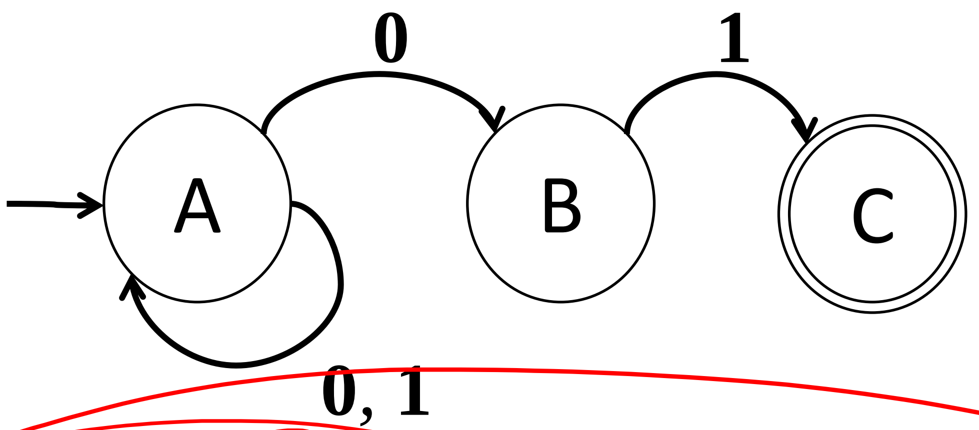
NFA–Comp $\subseteq$ DFA–Comp



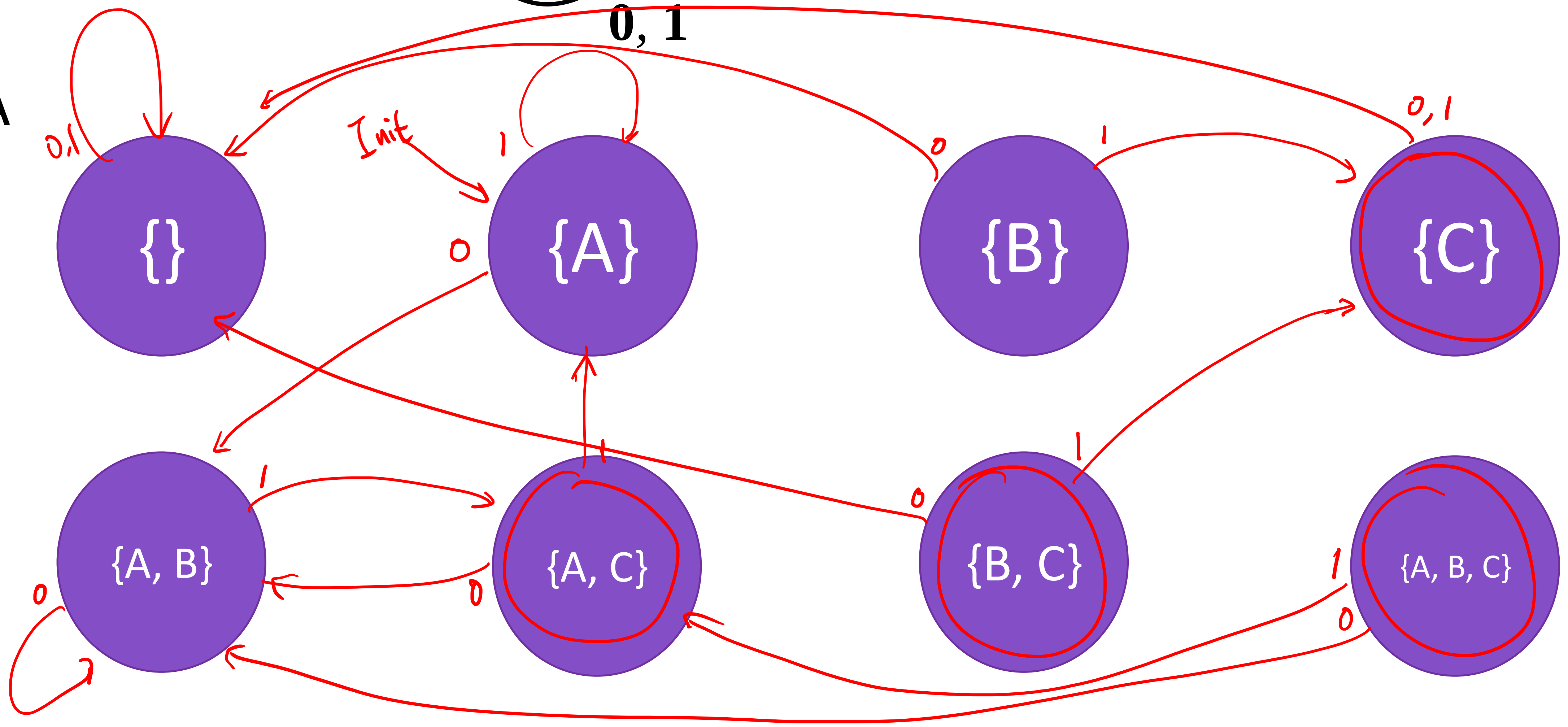
Can we construct a DFA that computes the same function as this NFA?

Idea: the states of DFA is the power set of NFA

NFA



DFA



(C, T, S)

From NFA to DFA

A **Nondeterministic *Finite Automaton*** over alphabet $\{0,1\}$ is a tuple (C, T, S) where:

1. C --- the number of *states*
2. $T: [C] \times \{0,1\} \to pow([C])$ a transition function
3. $S \subseteq [C]$ --- the set of accept states

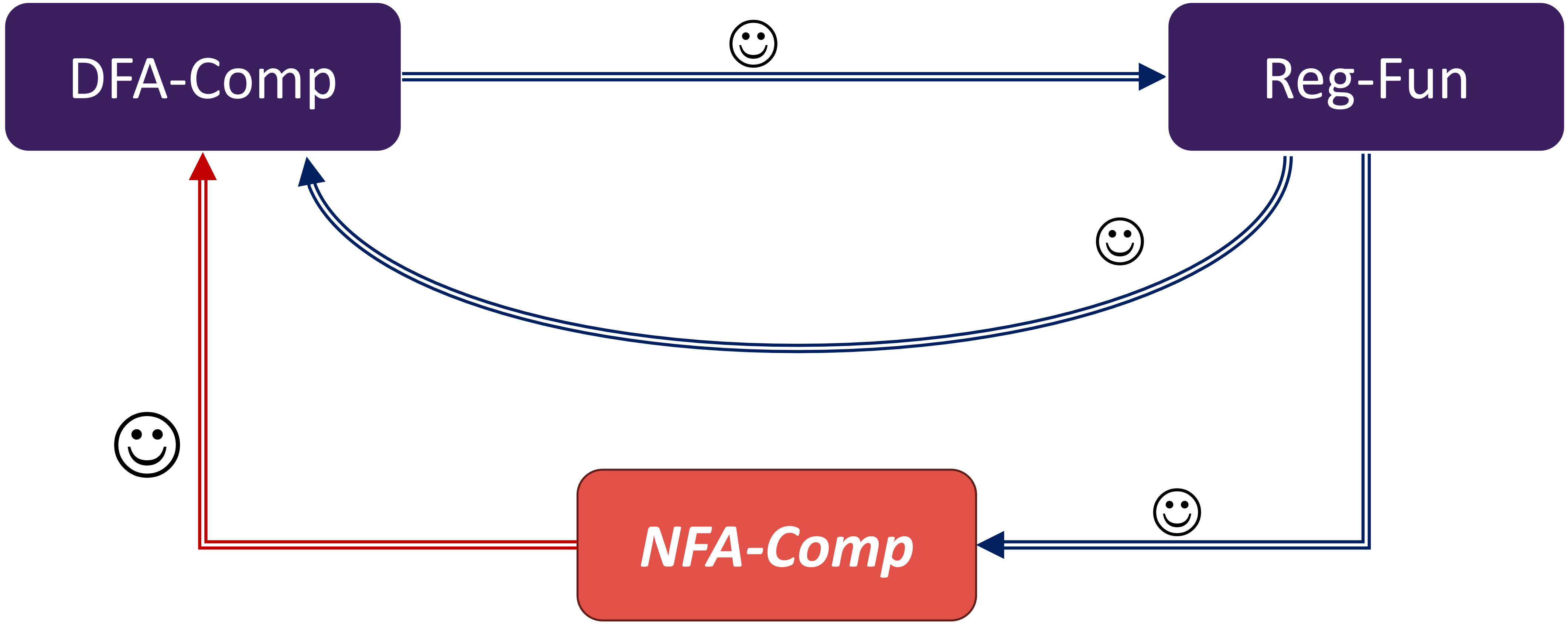
- States $[C'] = pow([C])$
- Init state $q_0' = \{q_0\}$
- Transition $T'(A \subseteq [C], b \in \{0,1\}) =$

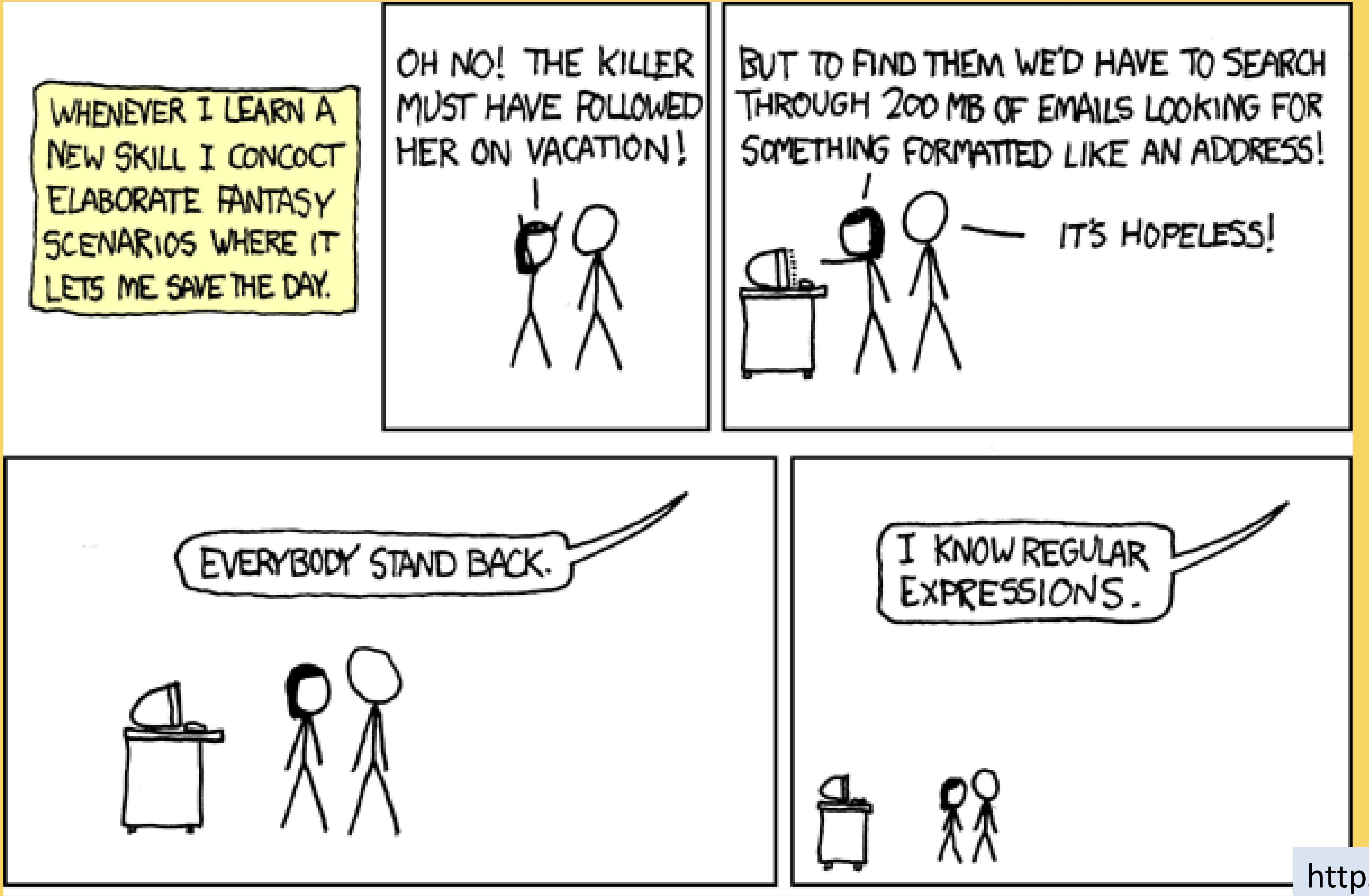
$$\bigcup_{s \in A} T(s, b)$$

$T'(A, b)$:
result $= \phi$
for all $s \in A$:
append $T(s, b)$ to result
return result

- Accept states $S' = \{A \subseteq [C]: \exists s \in S$ s.t. $s \in A\}$

High-Level Proof Plan





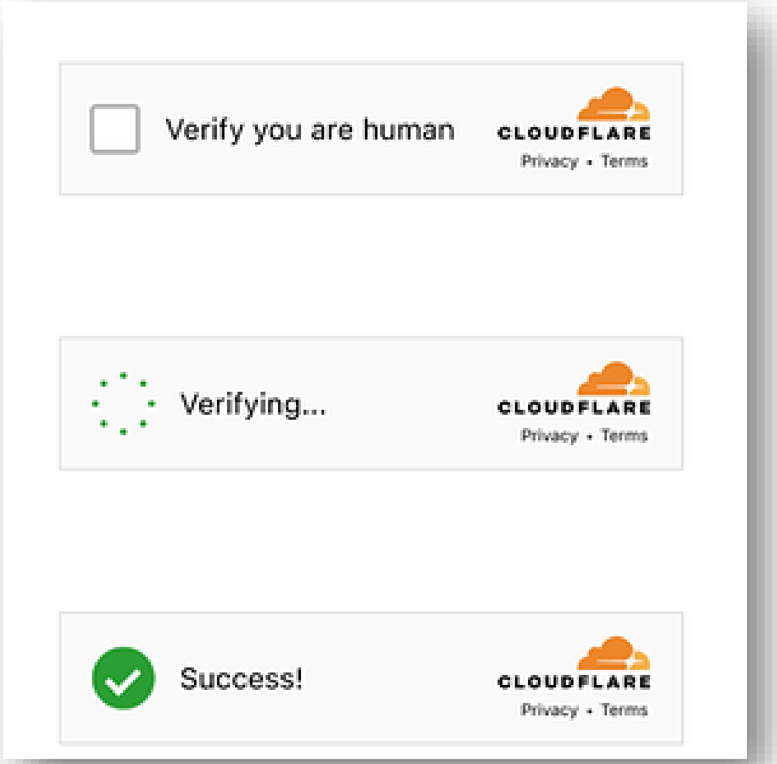
Regular Expressions in Practice

https://xkcd.com/208/

Cloudflare Outage

Cloudflare is an American technology company … provides a range of internet services, including content delivery network (CDN) services, cloud cybersecurity, DDoS mitigation, and ICANN-accredited domain registration.

The company's services act primarily as a reverse proxy between website visitors and a customer's hosting provider, improving performance and protecting against malicious traffic.
https://en.wikipedia.org/wiki/Cloudflare



https://community.weweb.io/t/tutorial-how-to-integrate-cloudflare-turnstile-google-recaptcha-alternative/3651

Details of the Cloudflare outage on July 2, 2019

2019-07-12

John Graham-Cumming

16 min read

https://blog.cloudflare.com/details-of-the-cloudflare-outage-on-july-2-2019/

The events of July 2

On July 2, we deployed a new rule in our WAF Managed Rules that caused CPUs to become exhausted on every CPU core that handles HTTP/HTTPS traffic on the Cloudflare network worldwide. We are constantly improving WAF Managed Rules to respond to new vulnerabilities and threats. In May, for example, we used the speed with which we can update the WAF to push a rule to protect against a serious SharePoint vulnerability. Being able to deploy rules quickly and globally is a critical feature of our WAF.

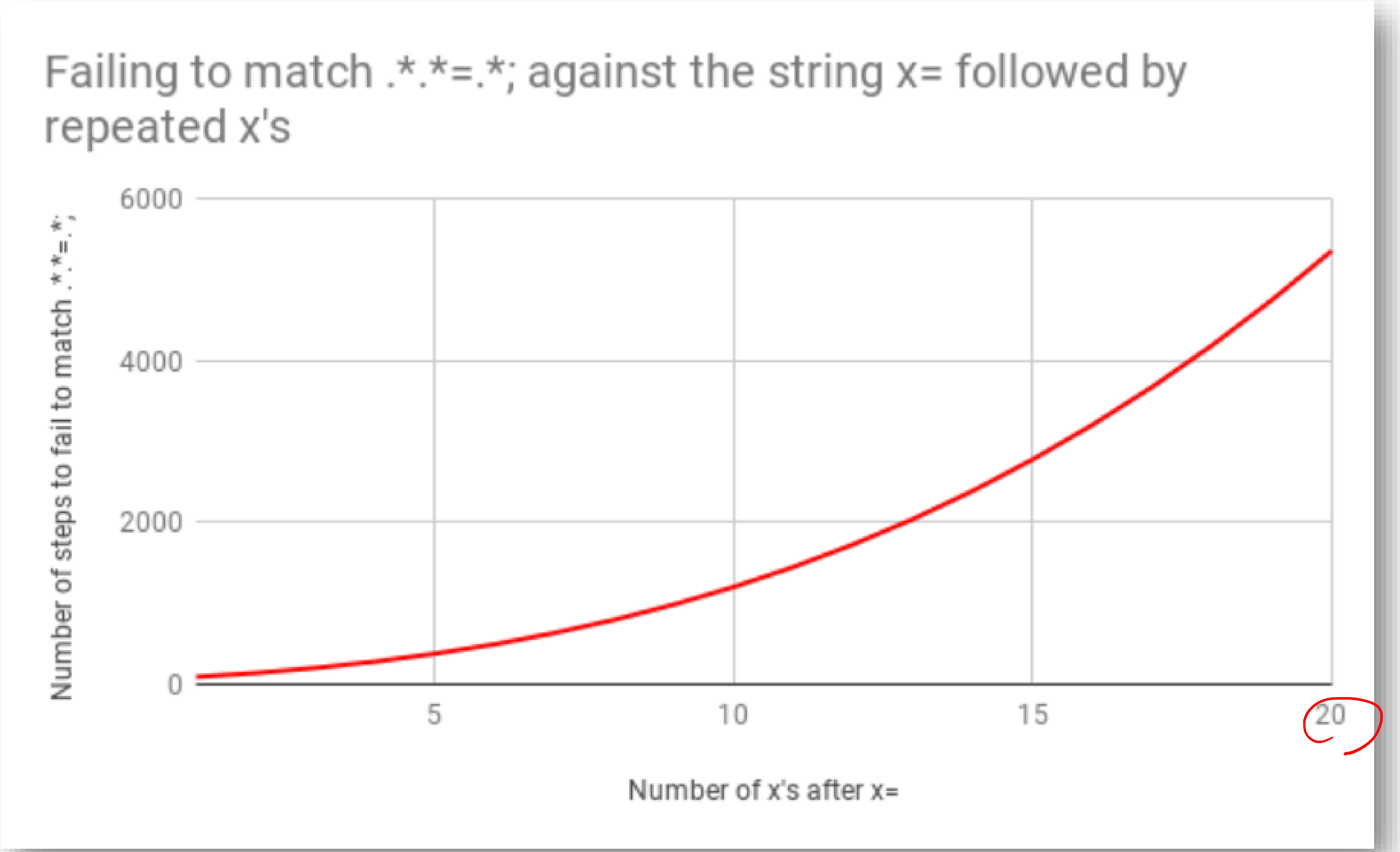
Unfortunately, last Tuesday's update contained a regular expression that backtracked enormously and exhausted CPU used for HTTP/HTTPS serving. This brought down Cloudflare's core proxying, CDN and WAF functionality. The following graph shows CPUs dedicated to serving HTTP/HTTPS traffic spiking to nearly 100% usage across the servers in our network.

Appendix: About Regular Expression Backtracking

To fully understand how ``(?:(?:\"|'|\]|\}|\\|\d|(?:nan|infinity|true|false|null|undefined|symbol|math)|\`|\-|\+)+[)]*;?((?:\s|-|~|!|{}|\|\||\+)*.*(?:.*=.*)))`` caused CPU exhaustion you need to understand a little about how a standard regular expression engine works. The critical part is `.*(?:.*=.*)`. The `(?:` and matching `)` are a non-capturing group (i.e. the expression inside the parentheses is grouped together as a single expression).

$. \equiv (0|1)$

For the purposes of the discussion of why this pattern causes CPU exhaustion we can safely ignore it and treat the pattern as `.*.*=.*`. When reduced to this, the pattern obviously looks unnecessarily complex, but what's important is any "real-world" expression (like the complex ones in our WAF rules) that ask the engine to "match anything followed by anything" can lead to catastrophic backtracking. Here's why.


Failing to match .*.*=.*; against the string x= followed by repeated x's
6000
4000
2000
0
5
10
15
20
Number of steps to fail to match .*.*=.*;
Number of x's after x=

RE: .* .* = .*

String: x=xx

x=xxxxxx

x=xxxxxxxxxxxxxxxxxxxxxxxxxxx

So far: Module 0 and 1

Module 0

Defining things Precisely:

- Natural numbers
- Sets
- Cardinality
- Infinity
- Countability

Module 1

Computing “regular” functions

- Regular expressions
- DFA, NFA
- ➔ All are “models” of computation
- Prove “equivalence” between models
- Prove limitations

We define <u>computation models</u> and prove things

More powerful models?

Survey Questions:

How will this class be different from Prof. Pettit's section?

This course:
- Sets and cardinality
- Regular expressions and DFA
- Circuits
- Computability and Turing machines
- Complexity and NP vs P

Prof. Pettit's:
- Sets and cardinality
- Regular expressions and DFA
- Context-free grammar
- Computability and Turing machines
- Complexity and NP vs P

Module 2: Circuits

A simple model of computation

Based on Boolean logical 'gates':

OR(a, b): outputs 1 iff a=1 or b=1

AND(a, b): outputs 1 iff a=1 and b=1

NOT(b): outputs 1 iff b=0

Output 0 otherwise

Plan

Universal circuits

Not universal gates

Circuit size hierarchy

[TCS] Textbook, Section 3 to 4

- https://introtcs.org/public/lec_03_computation.html

HW 3 due this Friday, Feb 13 (10:00pm)
HW 4 coming soon

Quiz 4 due next Mon, Feb 16