HW 2 due this Saturday, Feb 7 (10:00pm)
HW 3 coming soon

Quiz 3 due next Monday, Feb 9

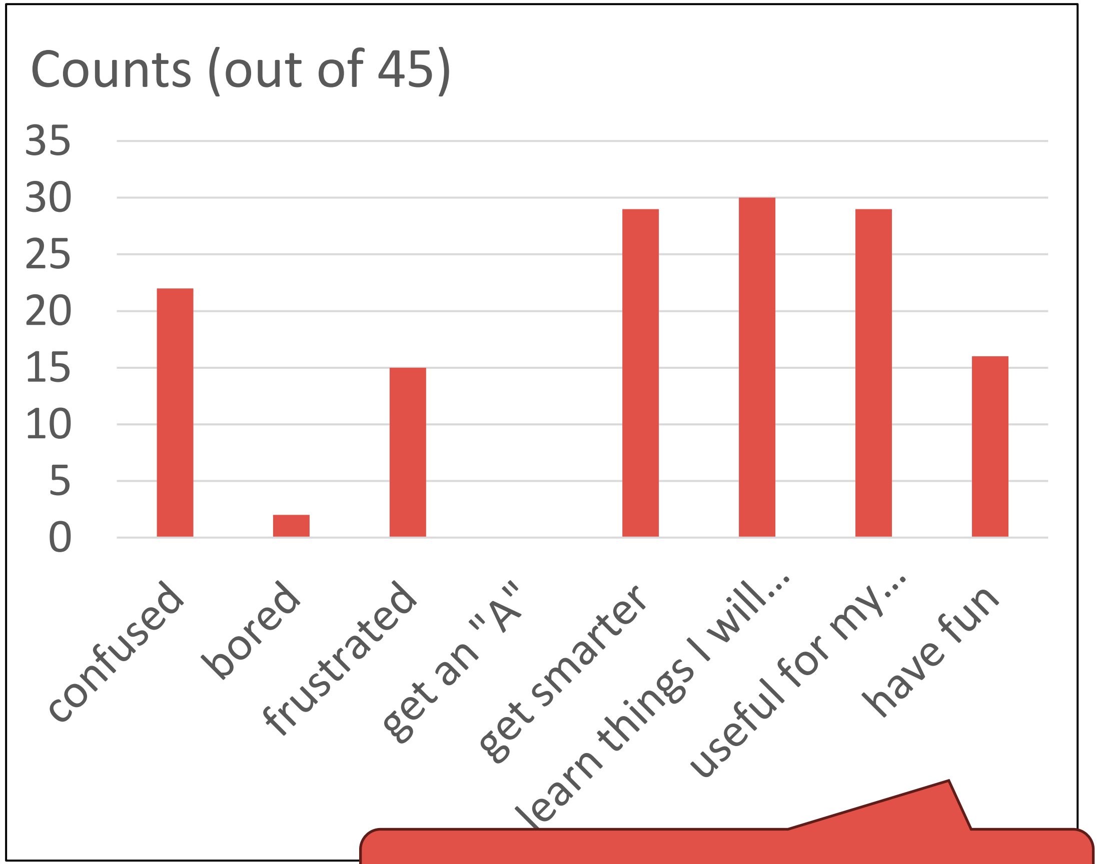
Photo: Haris Angelidakis

Class 7:
DFA ⇒ Reg Exp

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

Q2.3 What are you expecting from this class? (Check all that apply)
0 Points

- [ ] To be confused
- [ ] To be bored
- [ ] To be frustrated
- [ ] To get an "A"
- [ ] To get smarter
- [ ] To learn things I will find interesting
- [ ] To learn things that will be useful for my career goals
- [ ] To have fun



I hope we still have fun

Survey Questions

- “How would it compare in difficulty to the other major requirements?”
  “What are the ideas that everything else builds on?”

WK: each module consists of {a few definitions} and {a few theorems}, plus proofs and examples.
First-order logic is used everywhere.

- “LaTeX skill for assignments?”

WK: it makes graders’ lives easier.

Plan

Limitations

Pumping Lemma

Reg-Fun $\supseteq$ DFA-Comp

Reg-Fun $\subseteq$ DFA-Comp

Today: Chapter 6.4.2 in the TCS book

https://introtcs.org/public/lec_05_infinite.html#regdfaequivsec

Recap: Pumping Lemma

Theorem 6.21 (Pumping Lemma)

Then $\exists$

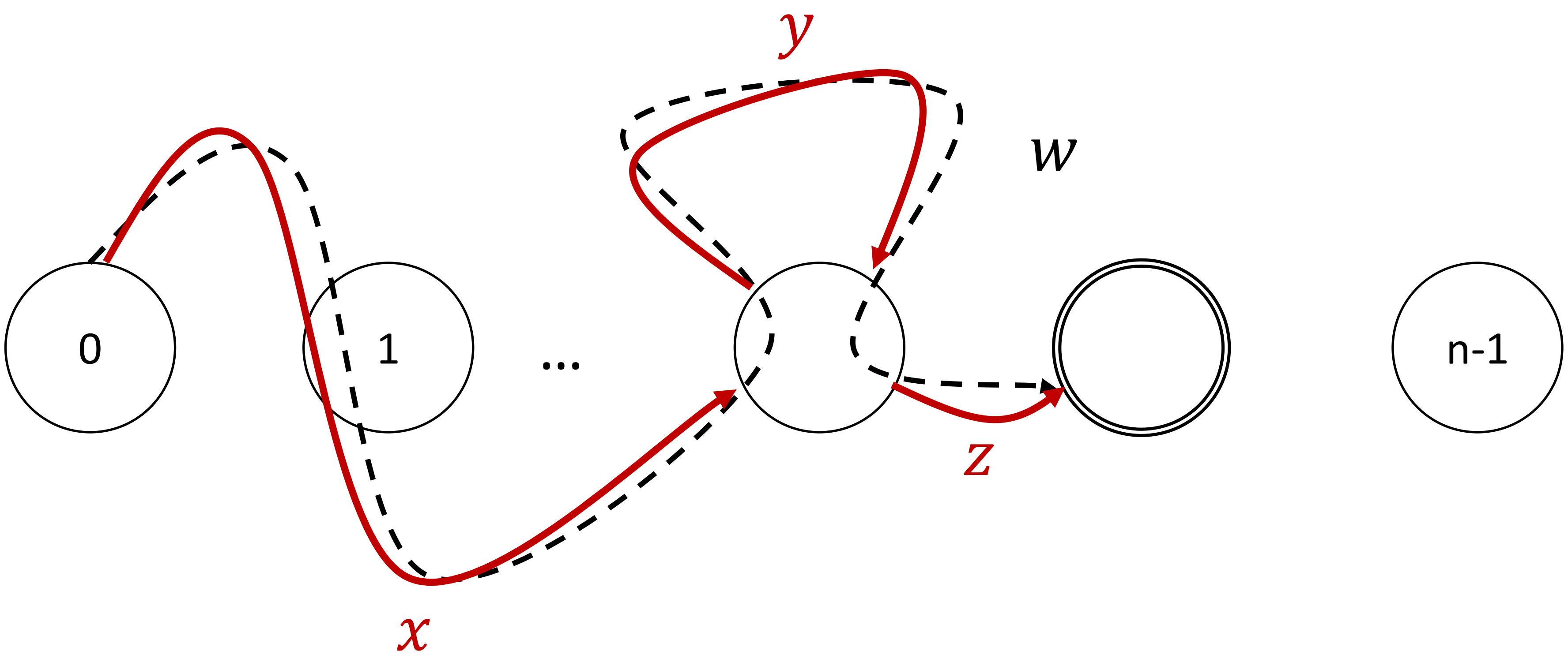
If *Let e be a regular expression over some alphabet Σ. Then there is some number n_0 such that for every $w \in \Sigma^*$ with $|w| > n_0$ and $\Phi_e(w) = 1$, we can write $w = xyz$ for strings $x, y, z \in \Sigma^*$*

$\forall$ *satisfying the following conditions:* $\exists$

1. $|y| \geq 1$.
2. $|xy| \leq n_0$.
3. $\Phi_e(xy^kz) = 1$ for $\forall$ every $k \in \mathbb{N}$.

Pumping Lemma, Illustrated

If DFA accepts a sufficiently long w, then there exist $xyz = w$ such that y is a loop.



Use Pumping Lemma: Prove Impossibility

Recap: A = $\{0^k1^k : k \in \mathrm{N}\}$ is not regular

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^i1^i and $0^{i+j}1^i$.

Either both will be accepted,

Or both will be rejected.

Ex: Palindroms is not regular

"" "1" "10110 1"

Lemma: e is regular expression, then
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

$\exists$ $\forall$ $\exists$ $\forall$

A string x is palindrome iff x is identical to reverse(x).

Proof.

For any n_0, let $w = 1\,0^{n_0}\,0^{n_0}\,1$

Where $w = xyz$ for $x = 1\,0^{n_0-2}$ $y = 0$ $z = 0\,0^{n_0}\,1$

1 $0\,n_0\,1$ $0\,0^{n_0}\,1$

$k=0$: xy^0z is not Pal

Ex: Palindroms is not regular

Lemma: e is regular expression, then
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

A string x is palindrome iff x is identical to reverse(x).

Proof.

For any n_0, let $w = 0^{n_0}10^{n_0}$

Where $w = xyz$ for $x = 0^{l_1}$, $y = 0^{l_2}$, $z = 0^{n_0-l_1-l_2}10^{n_0}$

$l_1 + l_2 \leq n_0$ and $l_2 \geq 1$.

But exists $k = 0$ s.t. xy^kz is not palindrome

Bogus proof: (011)* is not regular?!

Lemma: e is regular expression, then
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.

For any n_0, let $w = (011)^{n_0}(011)^{n_0}$

Where $w = xyz$ for $x = (011)^{n_0-1}01$, $y = 1$, $z = (011)^{n_0}$

But exists $k = 0$ s.t. xy^kz is not (011)*

Bogus proof:
(011)* is not regular?!

Lemma: e is regular expression, then
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.

1. For any n_0, let $w = (011)^{n_0}(011)^{n_0}$
2. Where $w = xyz$ for $x = (011)^{n_0-1}01$, $y = 1$, $z = (011)^{n_0}$
3. But exists $k = 0$ s.t. $xy^k z$ is not (011)*

some, we need all xyz s.t.

Must be wrong: (011)* is a regular expression!
Which line is the first wrong step?

Negation of Pumping Lemma

Theorem 6.21 (Pumping Lemma)

Then Not

If ~~Not~~ (

$\cancel{\exists}$ $\forall$ ~~Not~~

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

$\cancel{\forall}$ $\exists$ ~~Not~~

$\cancel{\exists}$ $\forall$ ~~Not~~

1. $|y| \geq 1$.
2. $|xy| \leq n_0$.
3. $\Phi_e(xy^kz) \neq 1$ for $\exists$ $\cancel{\text{every}}$ $k \in \mathbb{N}$.)

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

1, 11, 1111, 11111111, ...

Proof:

Given n_0

Lemma: Not e is regular expression, then
There is some number $\forall$ n_0 such that
for $\exists$ every $w \in \{0,1\}^*$ with $|w| > n_0$ and $\Phi_e(w) \; = \; 1$,
we can write $w = xyz$ for strings $\forall$ $x, y, z \in \{0,1\}^*$
satisfying the following conditions:

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

Why do we care regular or not?

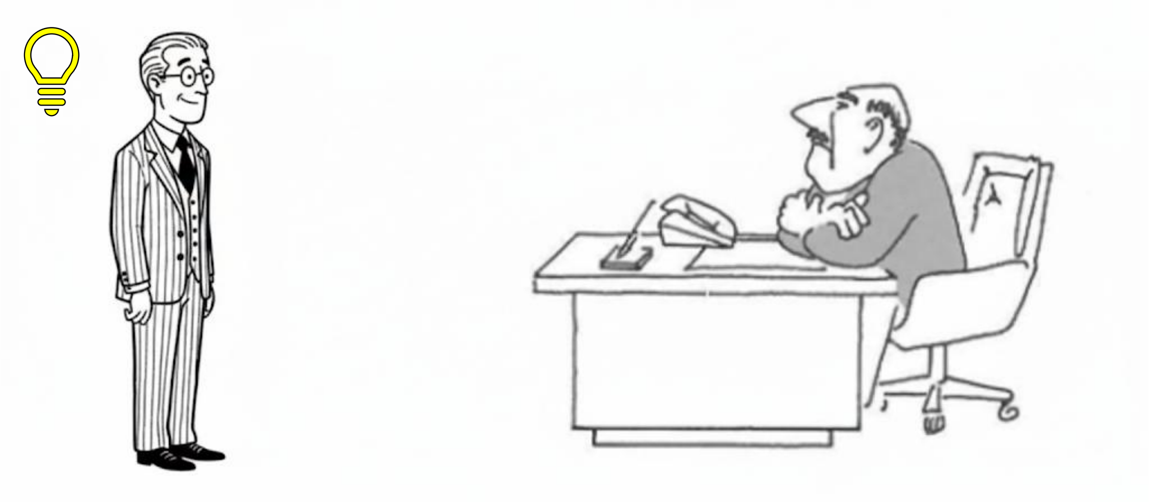
So far

Reg-Fun = DFA-Comp:
a good way to construct DFA and reg exp

Pumping Lemma:

to prove a function is not computable by any DFA

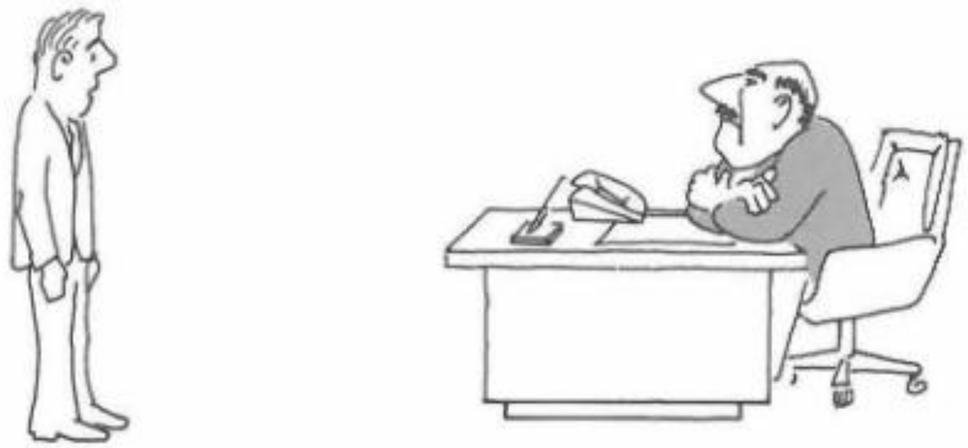
If there is a solution, we want it



Source: Computers and Intractability: A Guide to the Theory of NP-Completeness
Gemini modified.

What if there is no solution?

“Is that just me who couldn’t solve it?”



“I can’t find an efficient algorithm, I guess I’m just too dumb.”

Computers and Intractability: A Guide to the Theory of NP-Completeness
Michael R. Garey, David S. Johnson (1979)

If we can prove there is no solution

Much better!

"I can't find an efficient algorithm, because no such algorithm is possible!"

Use another tool (stronger than regular expression)

Proof of:
Reg-Fun $=$ DFA-Comp

Motivations of the Proof

- Practical needs:
  Writing regular expressions or its interpreter
  Writing DFAs (they are everywhere)

- Theoretical and Conceptual:
  Cool ideas in DFAs
  Manipulating CS abstractions (eg prog. languages)

- Learn “non-determinism” (stay tuned)

High Level Picture

Reg-Fun: the set of all regular functions.

DFA-Comp: the set $\{f \mid f \text{ is computed by some DFA } M\}$

Want:

1. DFA-Comp $\subseteq$ Reg-Fun,   and   2. Reg-Fun $\subseteq$ DFA-Comp

Hence, we need:

1. For each DFA M, there exists regular expression e such that for all x, $\Phi_e(x) = M(x)$
2. (The other direction)

Reg-Fun $\supseteq$ DFA-Comp

TCS, Section 6.4.2

For each DFA M, there is an equivalent regular expression e

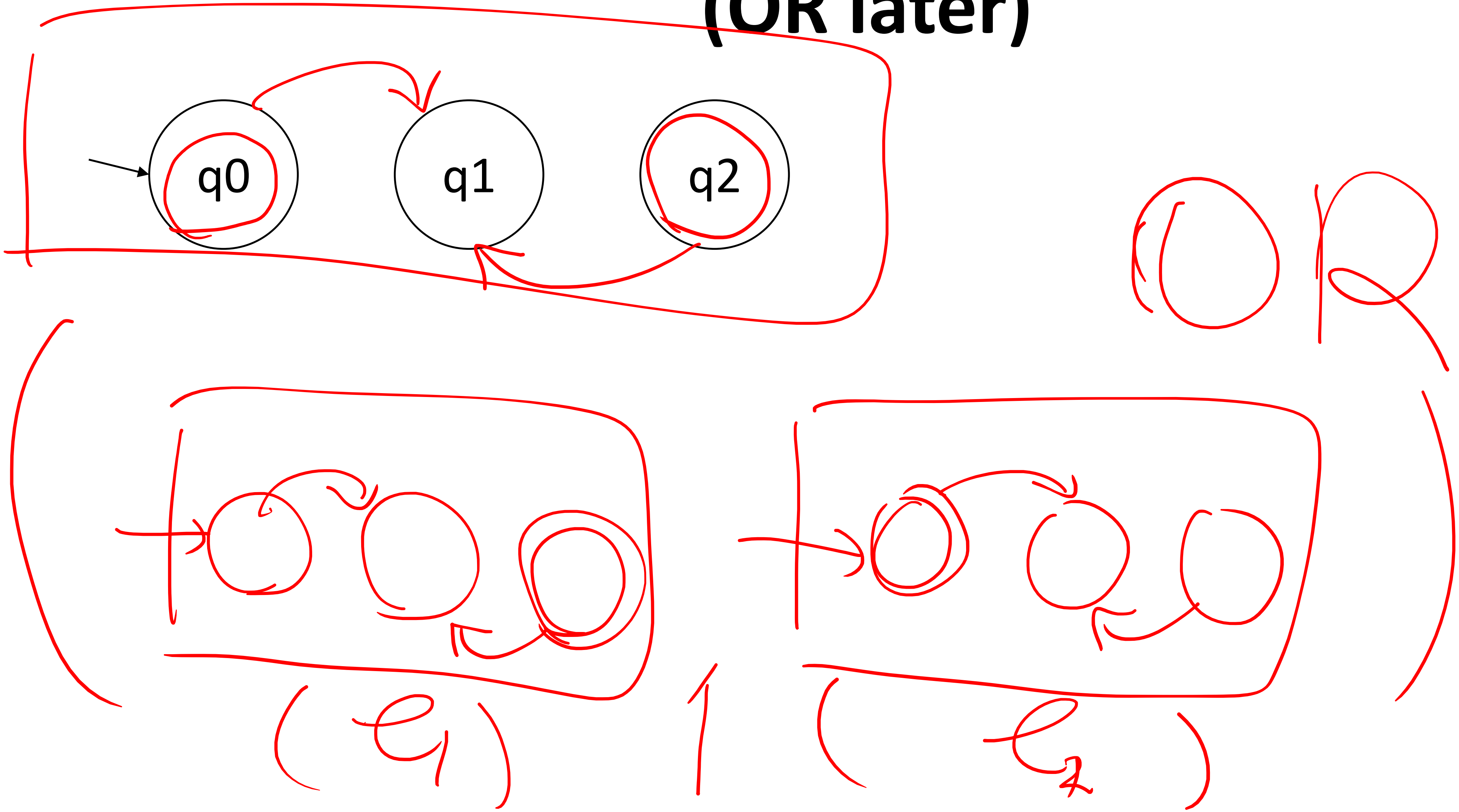
Proof? Generic way to construct such M for all e.

Algorithm: input e, output M

Proof idea 1: Consider only one Accept state (OR later)

Proof idea 2: Induction on subsets of states (see next

Proof idea 1: Consider only one Accept state (OR later)



Proof idea 2: Induction on subsets of states

For any M, let {0, 1, 2, …, $C-1$} = [C] be states in M,
let 0 be the initial and $C-1$ be the accept state.

Let t be natural num.

Consider the subset $[t] = \{0,1,\dots,t-1\}$.

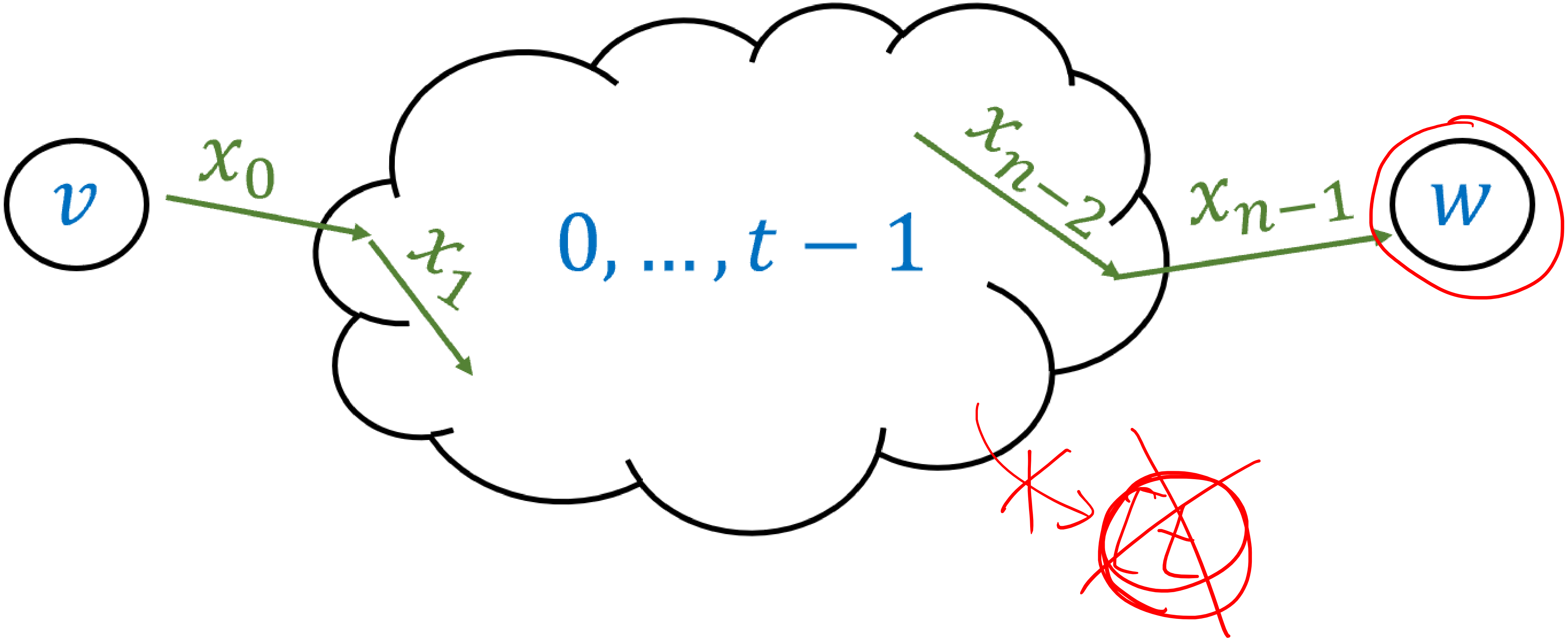
For any v, w,
let $R^t_{\boldsymbol{v},\boldsymbol{w}}$ be the strings go from $\boldsymbol{v}$ to $\boldsymbol{w}$ **only through** $[\boldsymbol{t}]$.

Consider the subset $[t] = \{0,1, \dots, t-1\}$. $\subseteq [C]$, $t \leq C$

Let $R_{v,w}^{t}$ be the strings go from v to w **only through nodes in** $[\boldsymbol{t}]$.

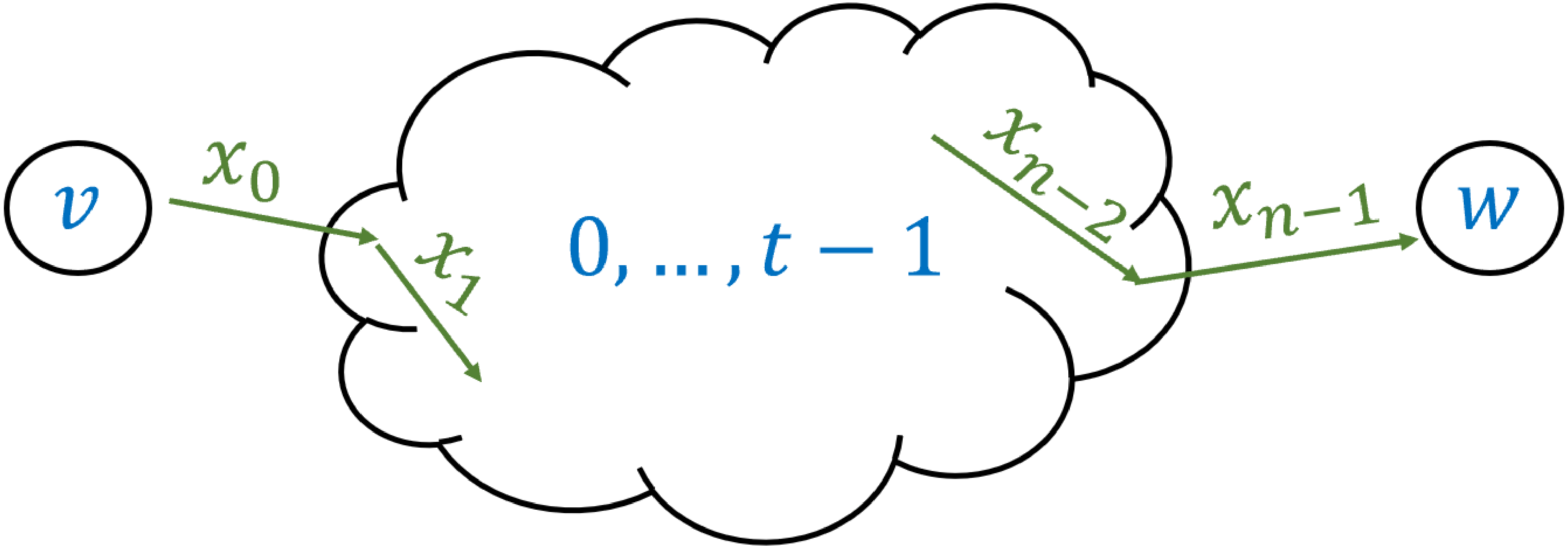
$v, w \in [C]$

Not even v or w if not in [t]



Induction predicate, P(t):
There exists a regular expression e_t matches $R^t_{v,w}$ for all v, w

$R^C_{v,w}$



Induction predicate, P(t): $t=0$ $[t] = \phi$
There exists a regular expression e_t matches $R_{v,w}^t$ for all v, w

Base case, P(0), [t] = emptyset

If $v = w$:

- 0 edge

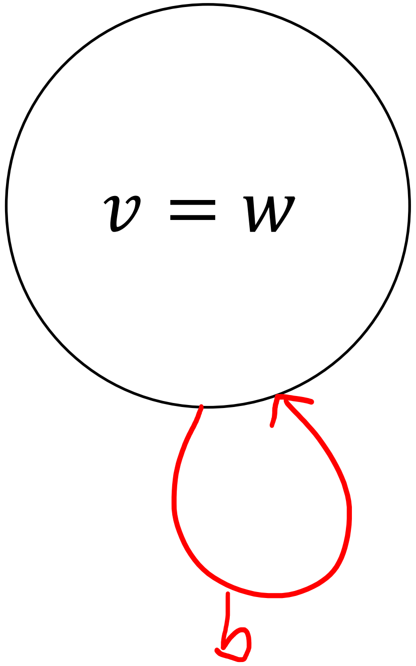
  e_0 : ϕ

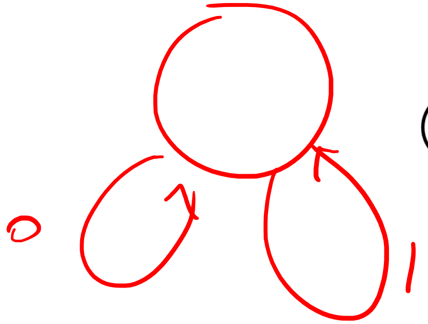
- 1 edge

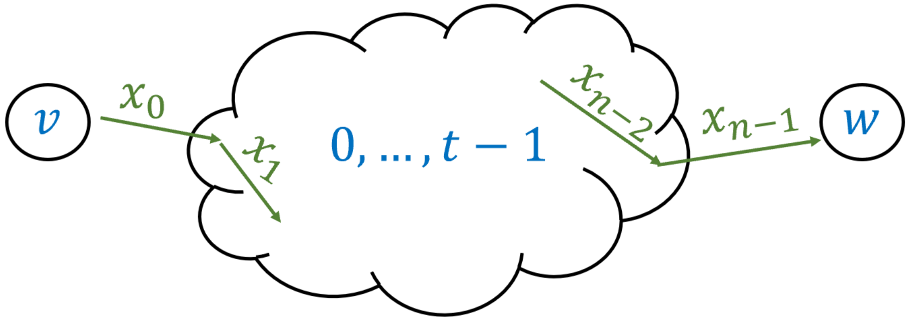
  e_0 : b , $b \in \{0, 1\}$

- 2 edges

  e_0 : $(0 | 1)$

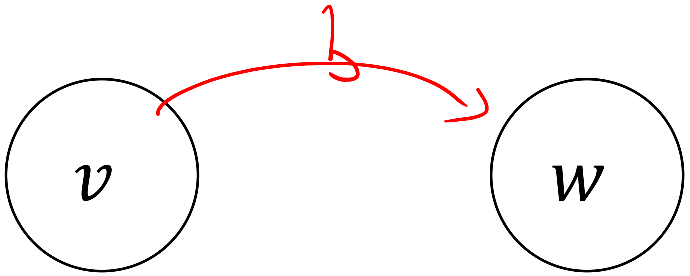






Induction predicate, P(t):
There exists a regular expression e_t matches $R^t_{v,w}$ for all v, w

Base case, P(0), [t] = emptyset



If $v \neq w$:

- 0 edge

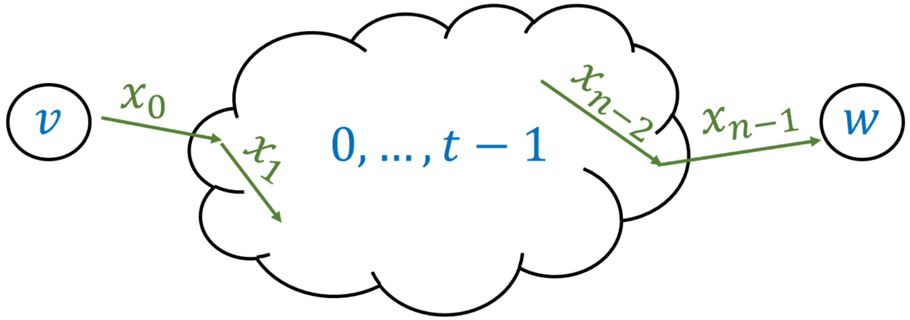
  e_0 : ϕ

- 1 edge

  e_0 : b

- 2 edges

  e_0 : (0 | 1)



Induction predicate, P(t):
There exists a regular expression e_t matches $R^t_{v,w}$ for all v, w

Inductive case, P(t) holds.

Want P(t+1): $R^{t+1}_{v,w}$ has a expression e_{t+1}

$$e_{t+1} =$$

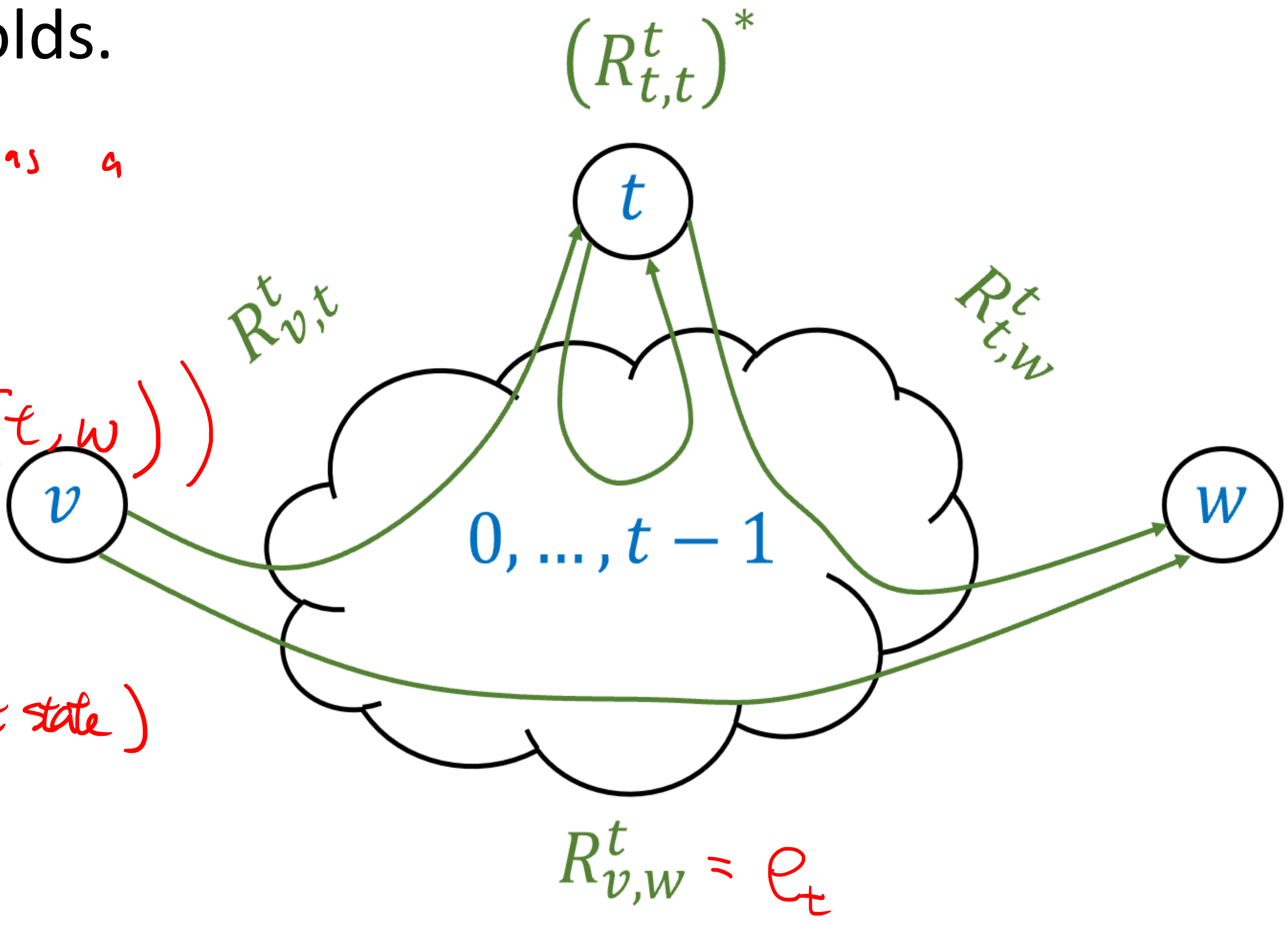
$$\big(\,(e_t[v,t])(e_t[t,t])^*(e_t[t,w])\,\big)$$

$$|\,(e_t \quad)$$

Comp (C, init state, accept state)

Compile(t, v, w):

recurse ....



DFA ⇒ Regular Expression

Application?

Next: Regular Expression ⇒ DFA

Logistics

- We are planning a makeup Midterm, aiming at late Feb. Need time and room, might not work for you. (We drop the lowest Midterm anyway.)
- Get free food/drink next Monday 10:30am at Greenberry's in Wilsdorf
- Survey: which office hours work for you?
  https://forms.cloud.microsoft/r/y5JuuYDbN9

HW 2 due this Saturday, Feb 7 (10:00pm)
HW 3 coming soon

Quiz 3 due next Monday, Feb 9