CSCI 361 Lecture 6: Pumping Lemma

Shikha Singh

Announcements & Logistics

- **HW 3** is due tomorrow **at 10 pm** on Gradescope
- Hand in Exercise #5, pick up Exercise #6
- Reminder: In class midterm 1 on Oct 7
- Looking ahead:
  - HW 4 will be released Thursday/ due Wed Oct 1
  - All content up to HW 4 is on the midterm
  - Practice midterm will be released Wed Oct 1
  - Use lecture time on Oct 2 will be review + practice problems

Last Time

- Discussed Myhill Nerode theorem
  - Max fooling set size of $L = \#$ equivalence classes of $\equiv_L$

    $=$ min states of DFA for L
- Practice using it to prove some languages are not regular
  - $\{a^n b^n | n \in \mathbb{N}\}$
  - $\{ww \mid w \in \{0,1\}^*\}$
  - $\{a^n \mid n \in \mathbb{N}$ and n is a power of 2$\}$
  - $\{w \in \{0,1\}^*$ has an equal number of 0s and 1s$\}$

Solution Sketches

- $L = \{a^i b^i \mid i \in \mathbb{N}\}$ is not regular
  - $F = \{a^i | i \in \mathbb{N}\}$ is an infinite fooling set since for any pair a^i, a^j $\in F$ such that $i \neq j$, suffix $z = b^i$ is such that $az \in L$ and $bz \notin L$
- $L = \{a^n \mid n \in \mathbb{N}$ and n is a power of $2\}$ is not regular
  - $F = \{a^{2^i} \mid i \in \mathbb{N}\}$ is an infinite fooling set
- $L = \{ww \mid w \in \{0,1\}^*\}$ is not regular
  - $F = \{0^i 1 \mid i \in \mathbb{N}\}$is an infinite fooling set

Two key properties: (a) **infinite** and (b) **pairwise distinguishable**

Review of Closure Properties

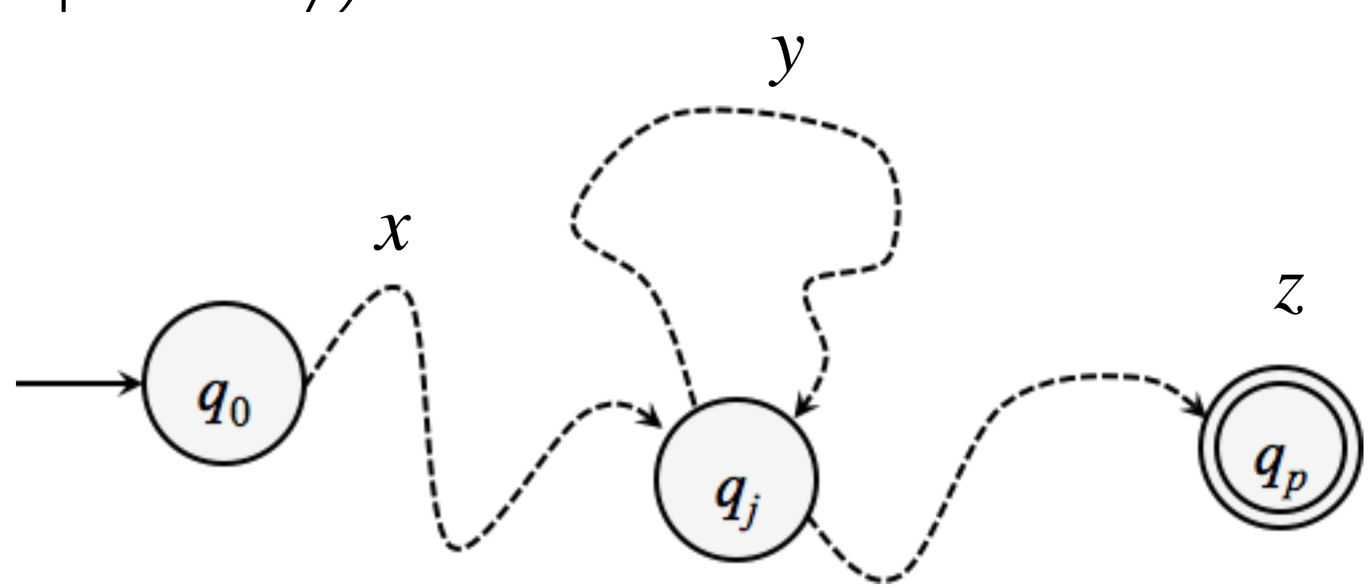
- Can often use closure properties to argue non-regularity
- $L = \{w \in \{0,1\}^*$ has an equal number of 0s and 1s$\}$ is not regular
- We know $L' = \{0^n 1^n \mid n \in \mathbb{N}_0\}$ is not regular (proved in class)
- Suppose L is regular, then since regular languages are closed under concatenation, we know $L \cap 0^*1^*$ must be regular
- But $L' = L \cap 0^*1^*$ which is a contradiction

Today: Pumping Lemma

- A necessary (but not sufficient) condition for regular languages
- Weaker guarantee but will generalize to the next class of languages

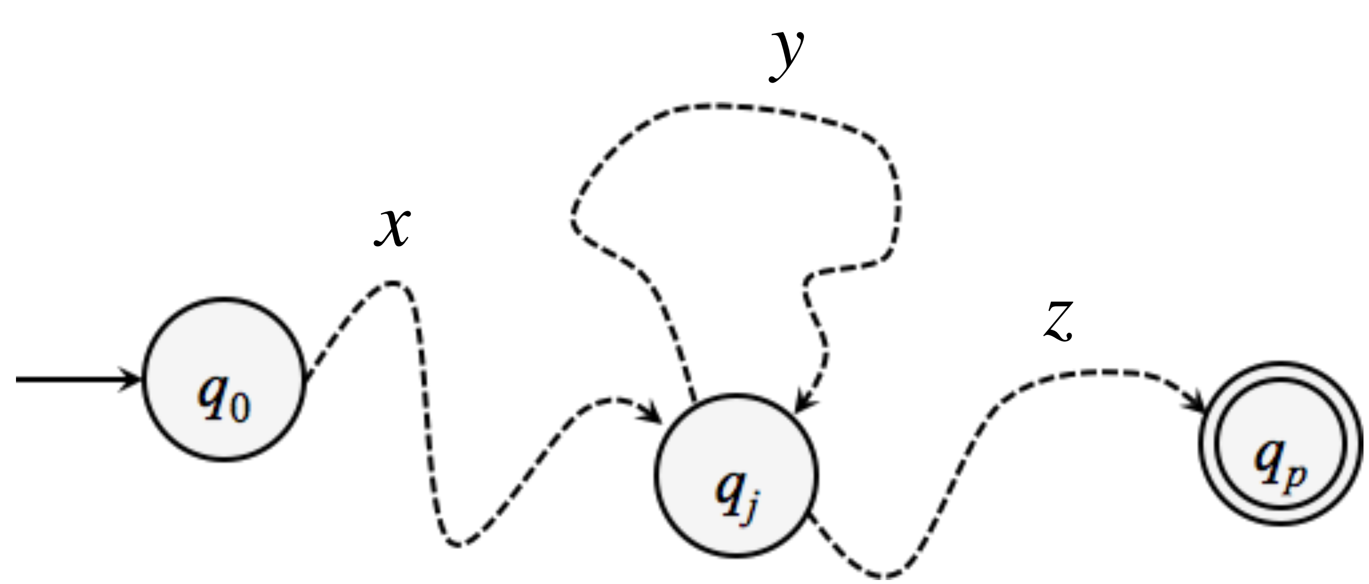
Pumping Lemma: Intuition

- If DFA M has p states then M visits a state more than once on any string with length at least p
  - Number of states visited = length of string $+ 1$
- Let $w = xyz$ be the string that is accepted such that y is component in between the first repeated state (q_j)
  - Then xy^iz should also be accepted (can "pump" the middle piece repeatedly)



Pumping Lemma: Proof

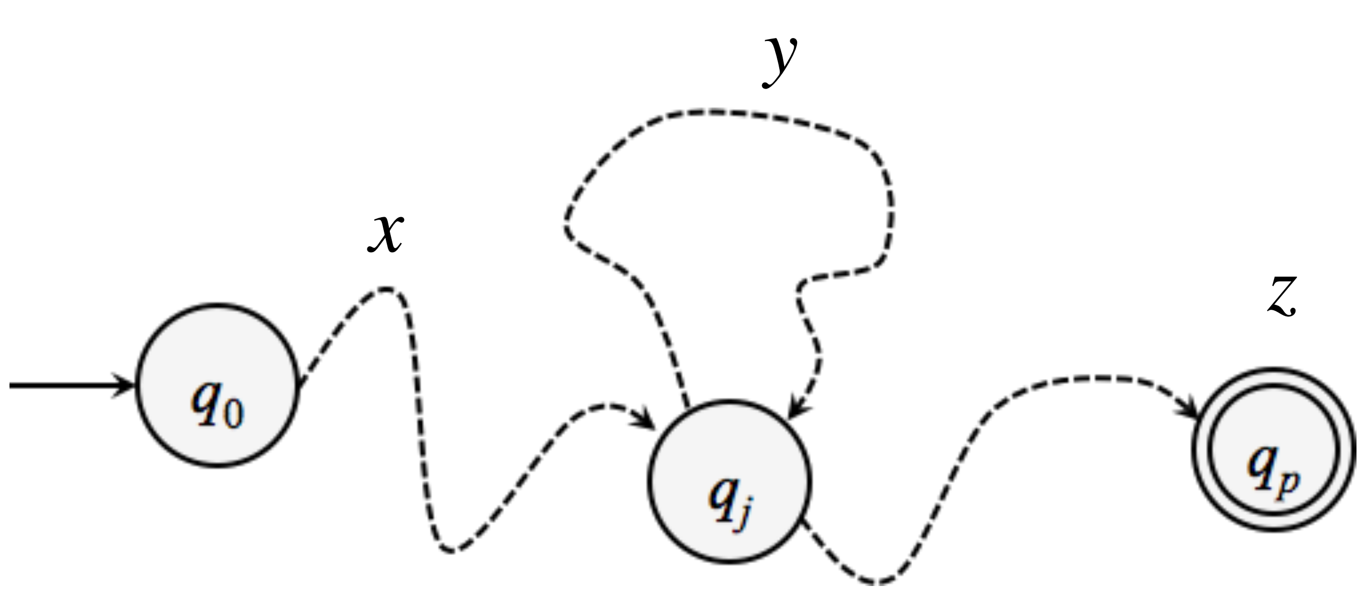
- Consider DFA M for L. Let p be the number of states in M
- Let s be a string of length $n \geq p$
- Then M's computation sequences enters $n + 1$ states on s
- By pigeonhole principle, there must be a repeated state q_j in the first $p + 1$ states of this sequence
- Let x be the substring that brings M from q_0 to first occurrence of q_j



Formal Statement

Pumping Lemma. If L is a regular language, then there exists a number p where if $w \in L$ is any string of length at least p, then w may be divided into three pieces $w = xyz$ such that:

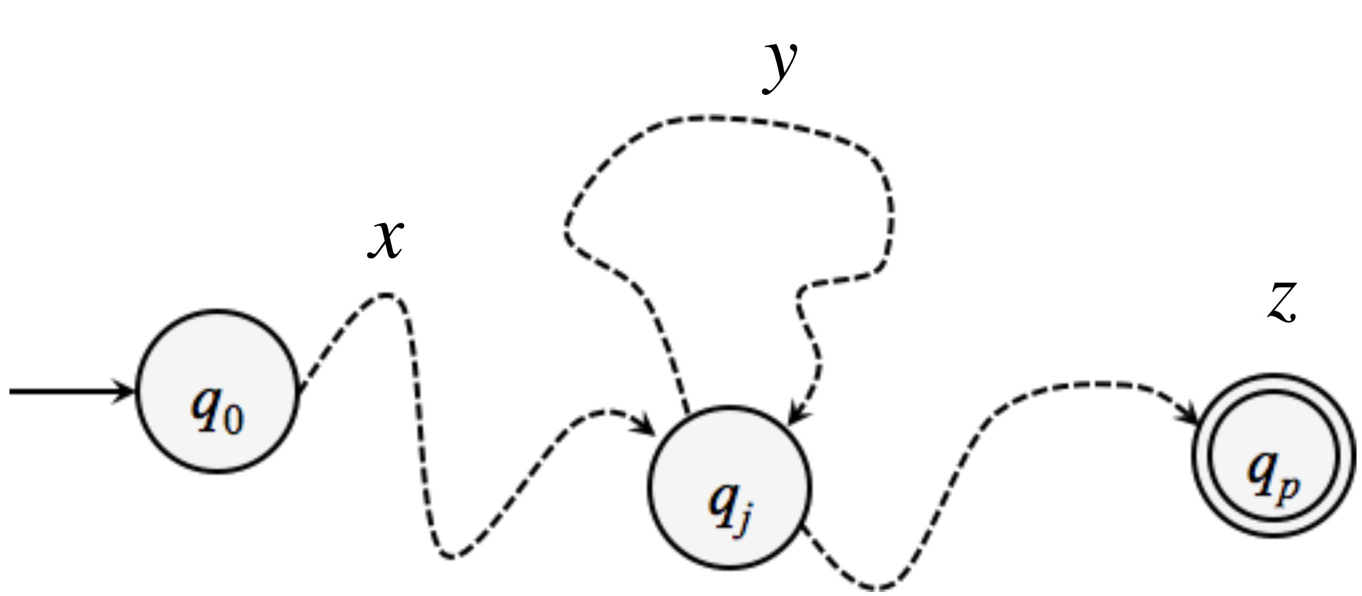
1. $|y| > 0$
2. $|xy| \leq p$ (y must appear amongst the first p symbols)
3. for each $i \geq 0$, $xy^iz \in L$



Formal Statement

Pumping Lemma. If L is a regular language, then **there exists** a number p such that **for all** $w \in L$ of length at least p, then **there exists** a way to divide w into three pieces $w = xyz$ such that:

1. $|y| > 0$
2. $|xy| \leq p$ (y must appear amongst the first p symbols)
3. for each $i \geq 0$, $xy^i z \in L$



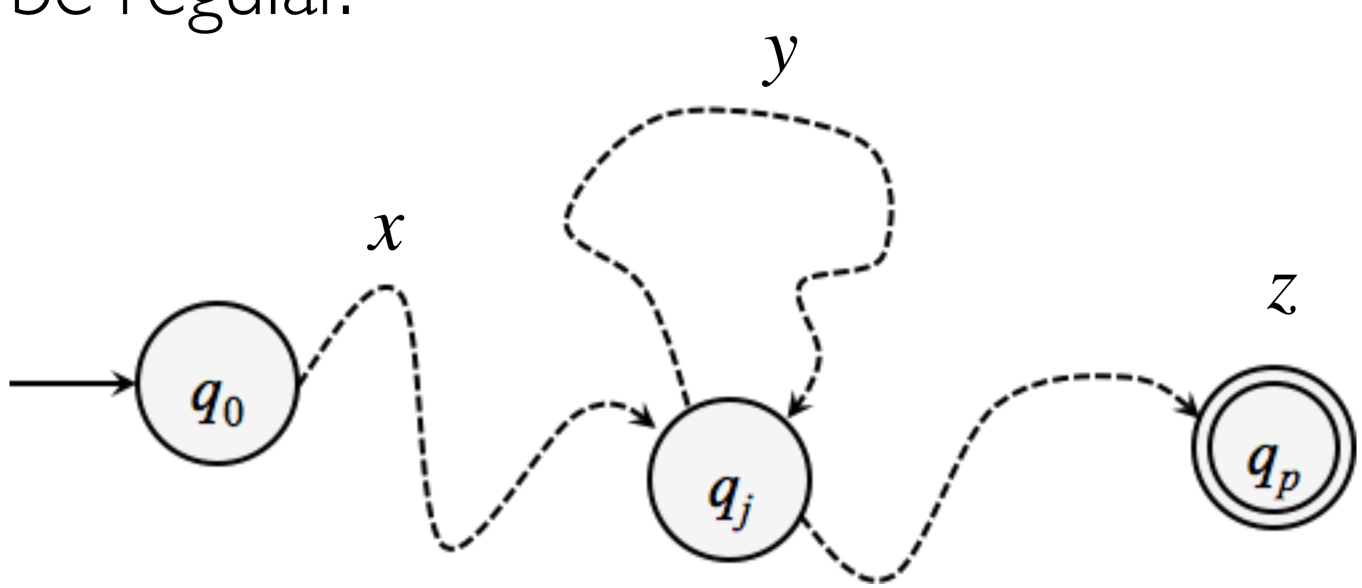
Equivalent Statement for Proofs

Pumping Lemma. If **for all** pumping lengths p **there exists** a string $w \in L$ of length at least p, such that **for all** possible ways to divide w into three pieces $w = xyz$ such that

(1) $|y| > 0$ and (2) $|xy| \leq p$ hold, it is possible to find an $i \geq 0$

for which (3) fails, i.e., $xy^iz \notin L$

then L cannot be regular.



Pumping Lemma: Game View

- Defender defends the claim that L satisfies pumping lemma
- Challenger challenges that claim and wants to show L does not satisfy pumping lemma

Defender		**Challenger**
Pick pumping length p	$\xrightarrow{p}$	
	$\xleftarrow{z}$	Pick $s \in L$ s.t. $\lvert s \rvert \geq p$
Divide s into xyz s.t. $\lvert y \rvert > 0$ and $\lvert xy \rvert \leq p$	$\xrightarrow{x, y, z}$	
	$\xleftarrow{i}$	Pick i, such that $xy^iz \notin L$

Pumping Lemma: Game View

- If L is regular: defender has a winning strategy, challenger gets stuck
- If challenger has a winning strategy, L cannot be regular

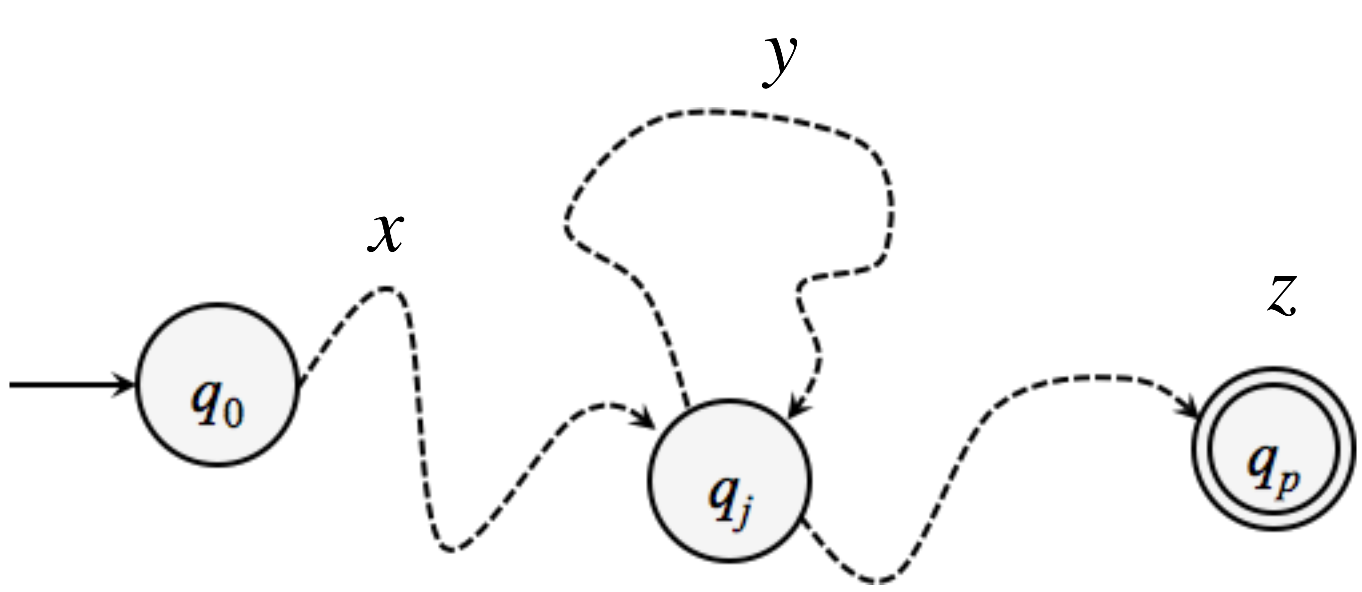
Defender		**Challenger**
Pick pumping length p	$\xrightarrow{p}$	
	$\xleftarrow{z}$	Pick $s \in L$ s.t. $\lvert s \rvert \geq p$
Divide s into xyz s.t. $\lvert y \rvert > 0$ and $\lvert xy \rvert \leq p$	$\xrightarrow{x, y, z}$	
	$\xleftarrow{i}$	Pick i, such that $xy^iz \notin L$

Questions

- Do all regular languages satisfy the pumping lemma?
- If a language satisfies the pumping lemma, does that mean it is regular?

Pumping Lemma Proof

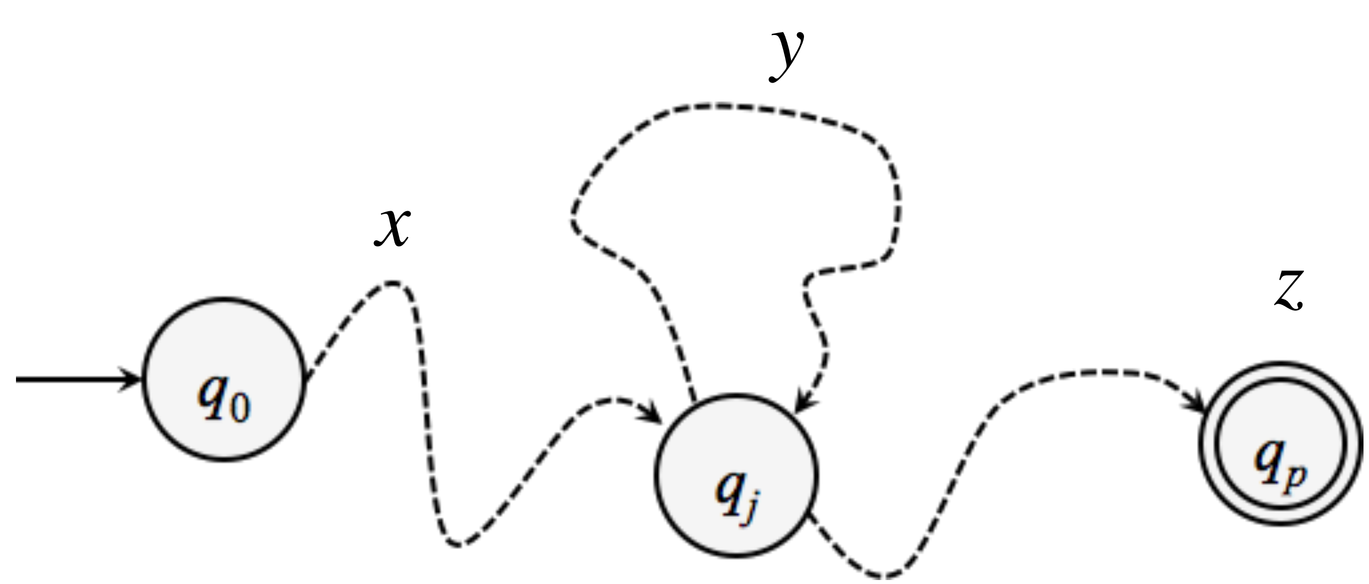
Proof. Let DFA M for L have p states. Let $w = w_1 \cdots w_n$ such that $n \geq p$ and $q_0, q_1, \ldots, q_n$ be the states entered by M on w.



Pumping Lemma Proof

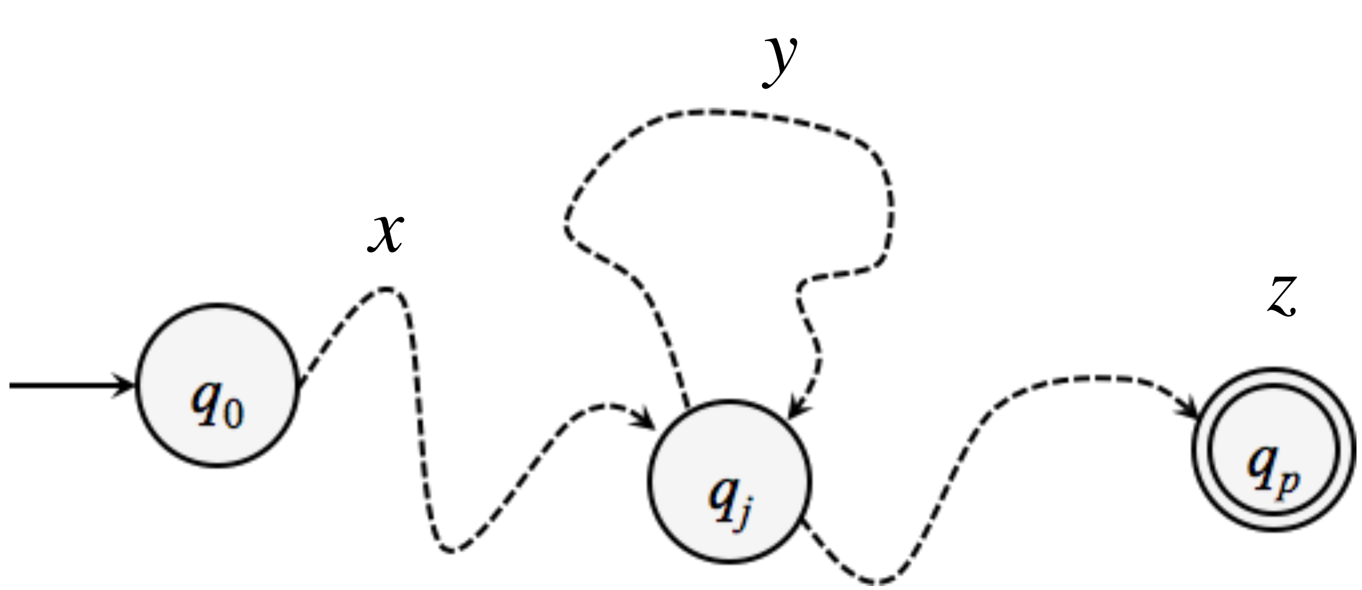
Proof. Let DFA M for L have p states. Let $w = w_1 \cdots w_n$ such that $n \geq p$ and $q_0, q_1, \ldots, q_n$ be the states entered by M on w. M must revisit a state in the first p symbols. Let q_j and q_k be the first and second occurrence of this state.

Let $x = w_1 w_2 \cdots w_{j-1}$, $y = w_j w_{j+1} \cdots w_k$ and $z = w_{k+1} \cdots w_n$ which satisfies the conditions (1) and (2). Condition (3) follows from the fact that the strings xy^i are all ***indistinguishable*** wrt M.



Pumping Lemma Proof

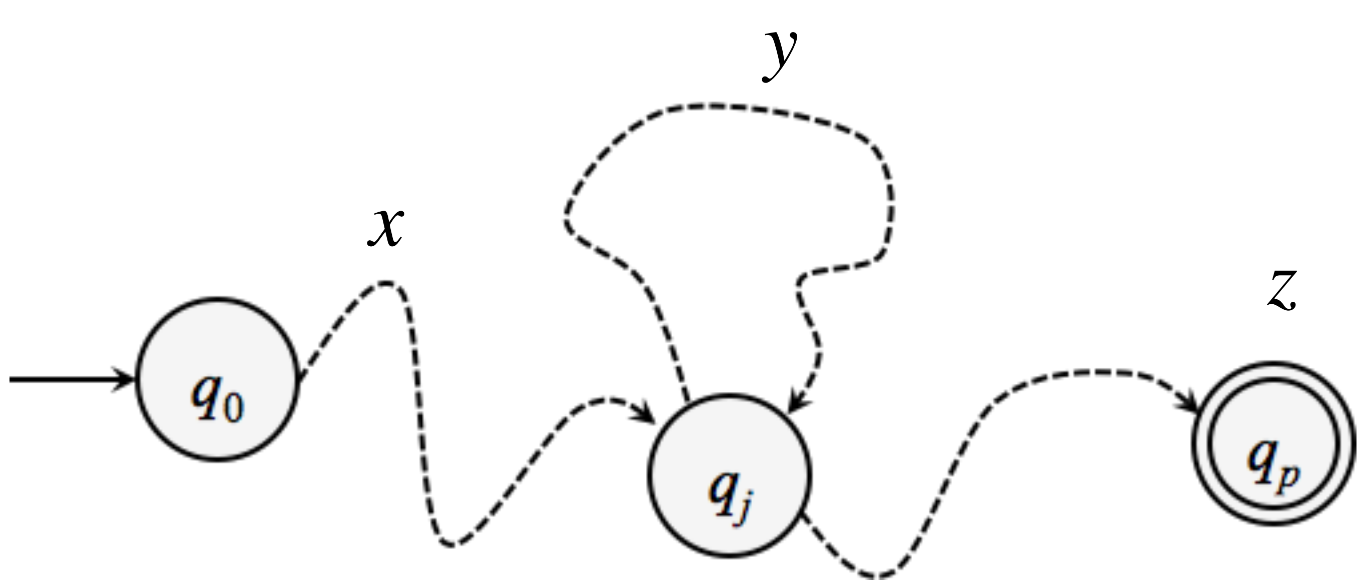
Proof. Let DFA M for L have p states. Let $w = w_1 \cdots w_n$ such that $n \geq p$ and $q_0, q_1, \ldots, q_n$ be the states entered by M on w. M must revisit a state in the first p symbols. Let q_j and q_k be the first and second occurrence of this state.



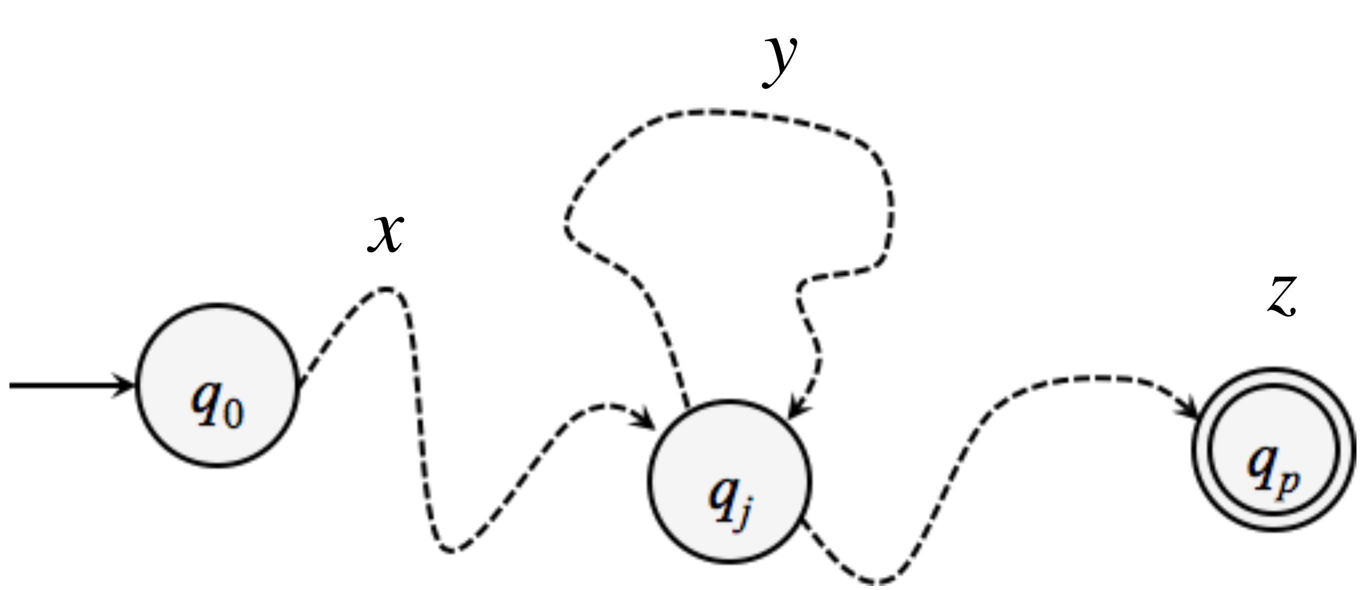
Pumping Lemma Proof

Proof. Let DFA M for L have p states. Let $w = w_1 \cdots w_n$ such that $n \geq p$ and $q_0, q_1, \ldots, q_n$ be the states entered by M on w. M must revisit a state in the first p symbols. Let q_j and q_k be the first and second occurrence of this state.

Let $x = w_1 w_2 \cdots w_{j-1}$, $y = w_j w_{j+1} \cdots w_k$ and $z = w_{k+1} \cdots w_n$ which satisfies the conditions (1) and (2). Condition (3) follows from the fact that the strings xy^i are all ***indistinguishable*** wrt M.




PUMP ALL THE STRINGS!
imgflip.com


y
x
z
q_0
q_j
q_p

Using Pumping Lemma Example

Problem 1. Prove that the language $L = \{0^n 1^{m+n} 1^n \mid m, n \geq 0\}$ is not regular using the pumping lemma.

Proof. Assume that L is regular. Let p be the pumping length.

Consider the string $w = 0^p 1^{2p} 0^p$. Then $w \in L$ and $|w| \geq p$.

Any way to divide w such that $|xy| \leq p$ and $|y| > 0$ means $y = 0^j$ for some $j \geq 1$, that is,

$x = 0^i$, $y = 0^j$ and $z = 0^{p-(i+j)} 1^{2p} 0^p$ where $i \geq 0, j \geq 1$

But then, $xz = 0^{p-j} 1^{2p} 0^p \notin L$ which is a contradiction $\Rightarrow \Leftarrow$

Review PL Steps

- Proving L is not regular using pumping lemma
  - Assume L is regular, let p be the pumping lemma given by lemma
  - Consider a specific string $w \in L$ of length at least p such that
  - for every possible partition of w into x, y, z satisfying
    - $|xp| \leq p$ and $|y| > 0$
  - there exists an i such that $xy^iz \notin L$
- The above steps provide a contradiction to L being regular by PL
- HW 4 Problem 5
  - Show that a language is not regular and show that it satisfies conditions of the pumping lemma

In Class Practice: Use Pumping Lemma

Problem 1. Prove that the language $L = \{a^i b^i \mid i \in \mathbb{N}\}$ is not regular.

Problem 2. Prove that $L = \{ww^R \mid w \in \{0,1\}^*\}$ is not regular.

Problem 3. Is the language $L = \{(ab)^i \circ (ab)^i \mid i \geq 0\}$ regular?

Problem 4. Prove that

$L = \{w \mid w \in \{0,1\}^*$ and the number of 1s in w is not equal to the number of 0s in $w\}$

is not regular.

Finite Automata Applications

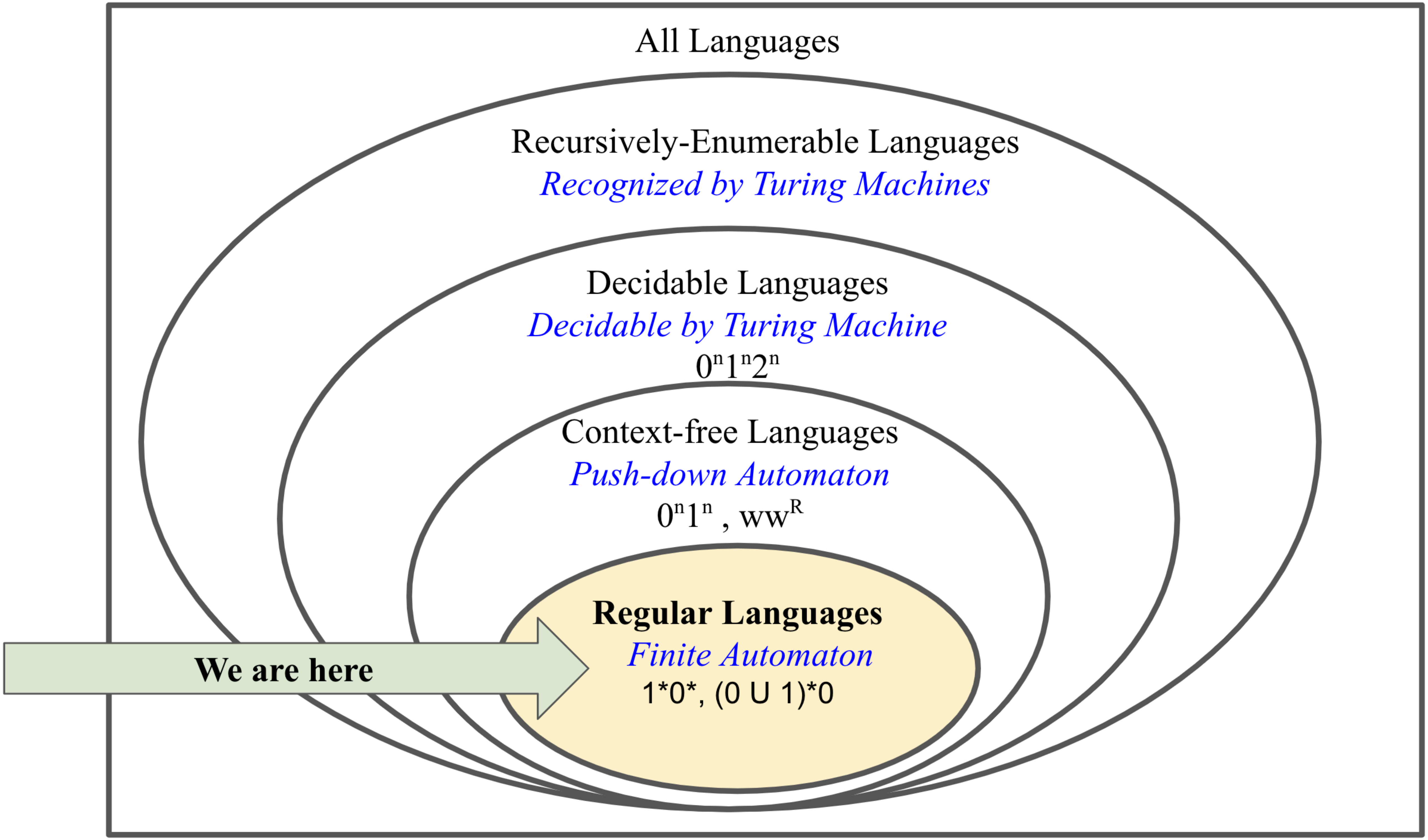
- Lexical analysis in compilers
- Networking protocols and routing
- Circuit design and event-driven programming
- Synchronization of distributed devices

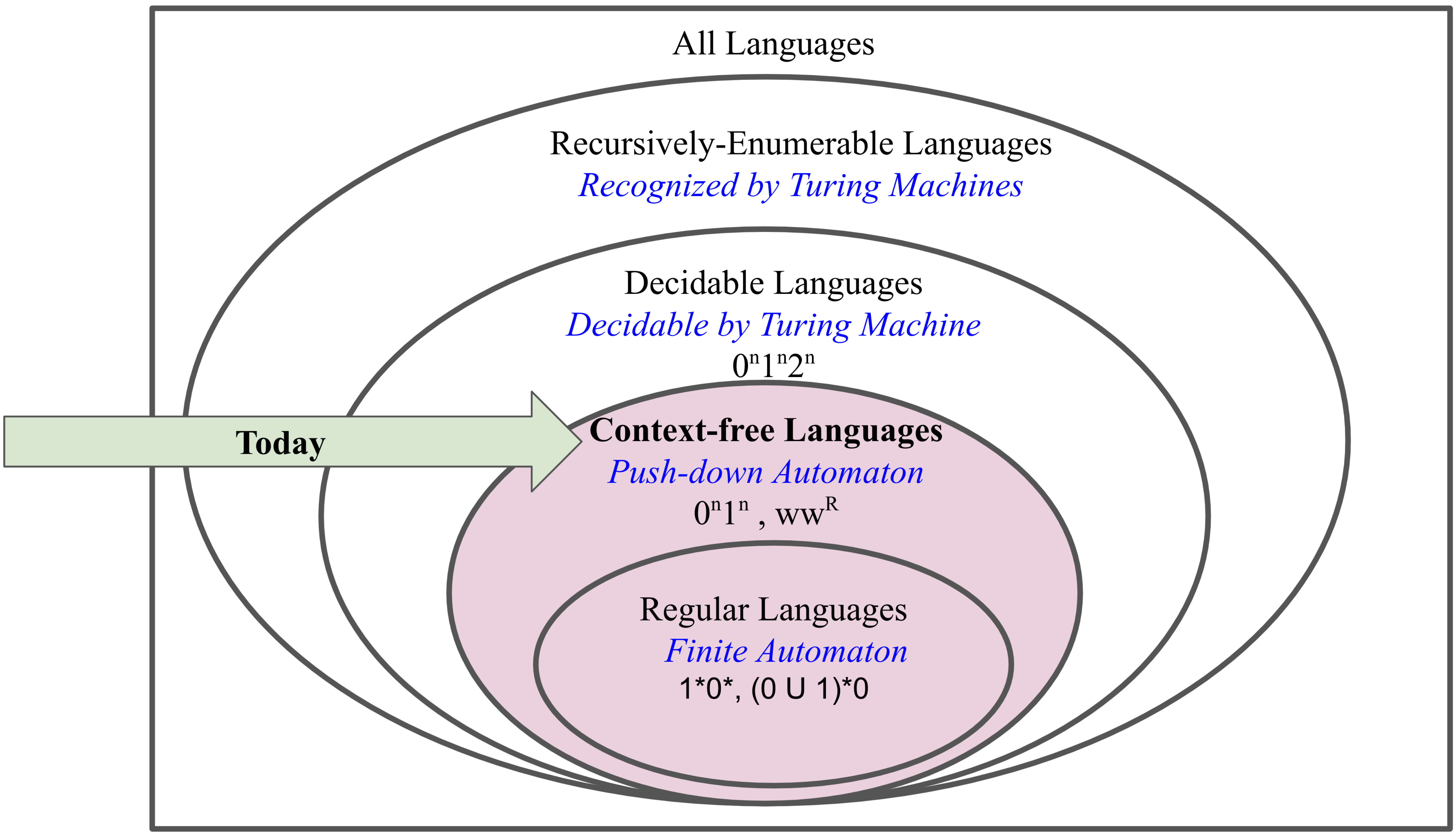
Finite Automata Limitation

- Code snippets that can only store finitely many states
- (Theory vs PL) Remember that all models of computation we discuss map directly to programming constructs

```
CONTAINS11(w[1..n]):
  found ← FALSE
  for i ← 1 to n
      if i = 1
          last2 ← w[1]
      else
          last2 ← w[i − 1] · w[i]
      if last2 = 11
          found ← TRUE
  return found
```

```
MULTIPLEOF5(w[1..n]):
  rem ← 0
  for i ← 1 to n
      rem ← (2 · rem + w[i]) mod 5
  if rem = 0
      return TRUE
  else
      return FALSE
```


All Languages
Recursively-Enumerable Languages
Recognized by Turing Machines
Decidable Languages
Decidable by Turing Machine
$0^n1^n2^n$
Context-free Languages
Push-down Automaton
0^n1^n , ww^R
Regular Languages
Finite Automaton
1*0*, (0 U 1)*0
We are here


All Languages
Recursively-Enumerable Languages
Recognized by Turing Machines
Decidable Languages
Decidable by Turing Machine
$0^n1^n2^n$
Today
Context-free Languages
Push-down Automaton
0^n1^n , ww^R
Regular Languages
Finite Automaton
1*0*, (0 U 1)*0