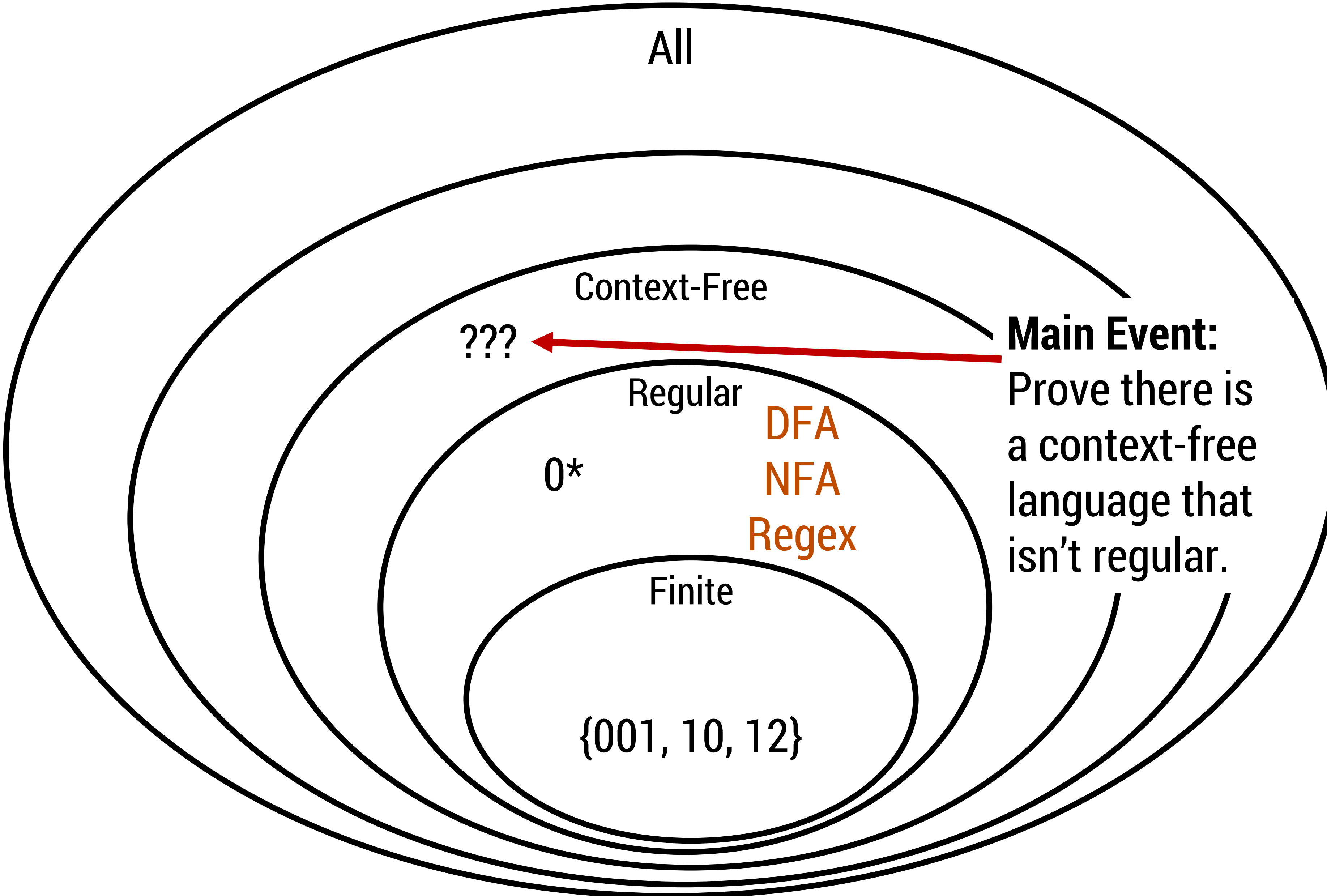
Fall 2015

Lecture 25: Limitations of DFAs (irregular languages)

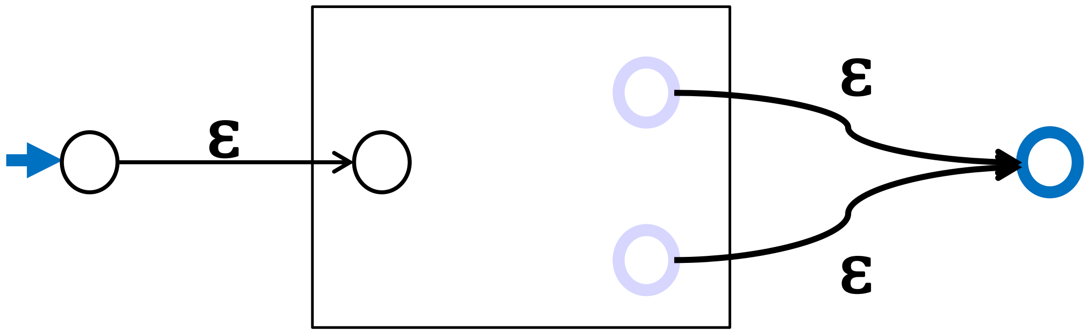

All
Context-Free
???
Regular
DFA
NFA
Regex
0*
Finite
{001, 10, 12}
Main Event:
Prove there is a context-free language that isn't regular.

generalized NFAs

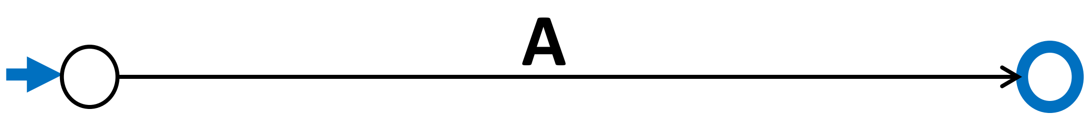
- Like NFAs but allow
  - Parallel edges
  - Regular Expressions as edge labels

    NFAs already have edges labeled **ε** or ***a***
- An edge labeled by **A** can be followed by reading a string of input chars that is in the language represented by **A**
- A string x is accepted iff there is a path from start to final state labeled by a regular expression whose language contains x

Add new start state and final state



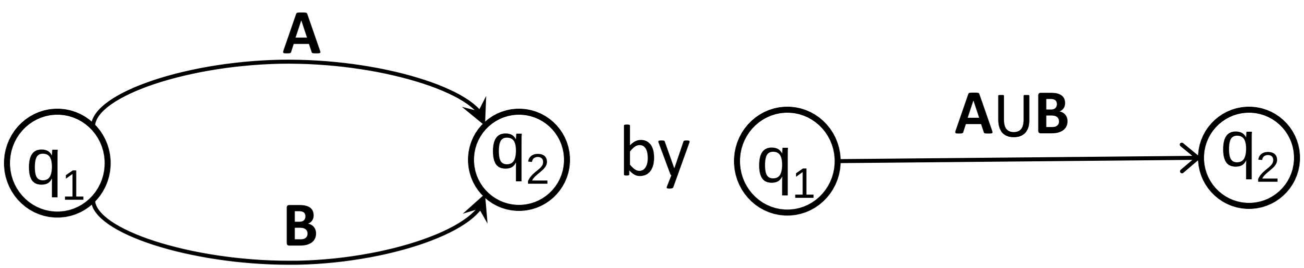
Then eliminate original states one by one, keeping the same language, until it looks like:



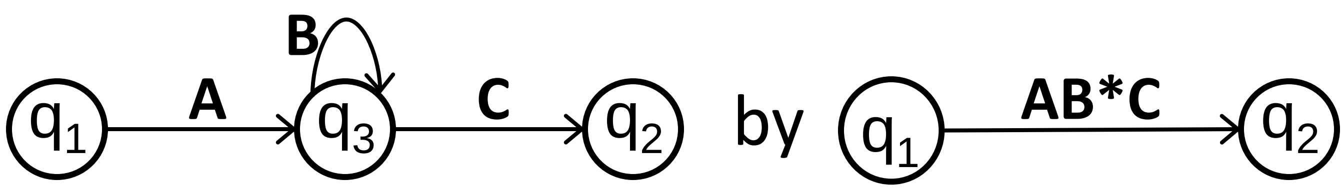
Final regular expression will be **A**

only two simplification rules

- Rule 1: For any two states q_1 and q_2 with parallel edges (possibly q_1=q_2), replace



- Rule 2: Eliminate non-start/final state q_3 by replacing all

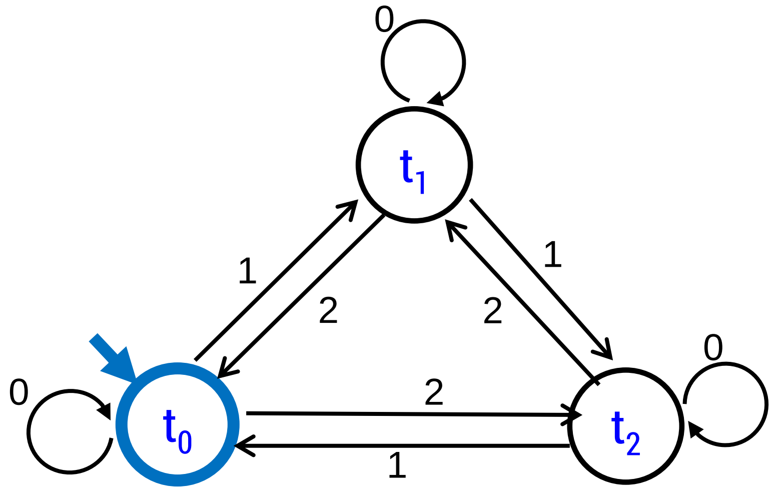


for *every* pair of states q_1, q_2 (even if q_1=q_2)

converting an NFA to a regular expression

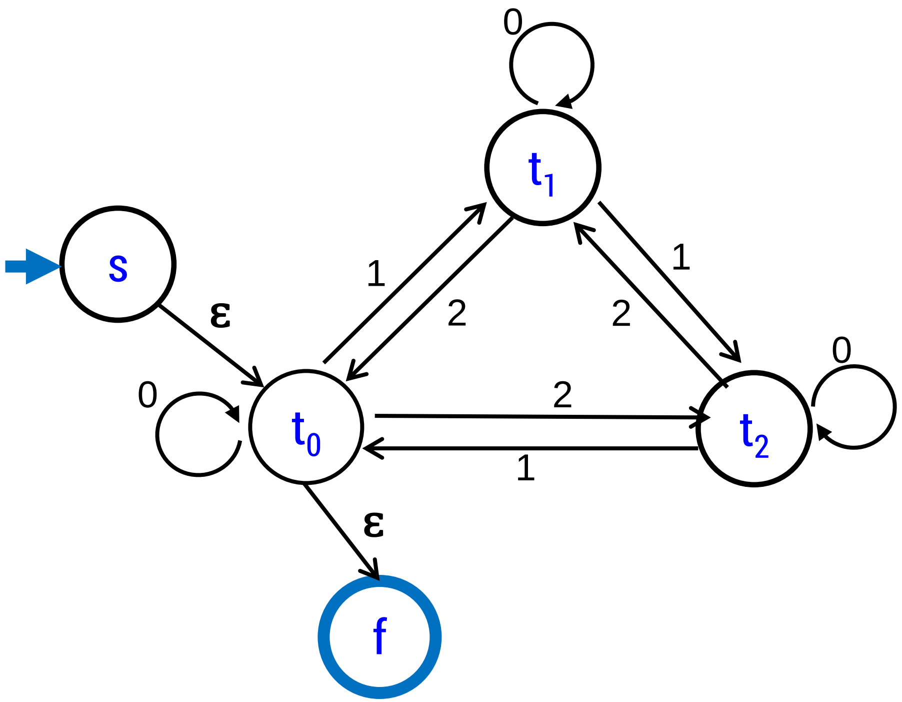
Consider the DFA for the mod 3 sum

- Accept strings from {0,1,2}* where the digits mod 3 sum of the digits is 0



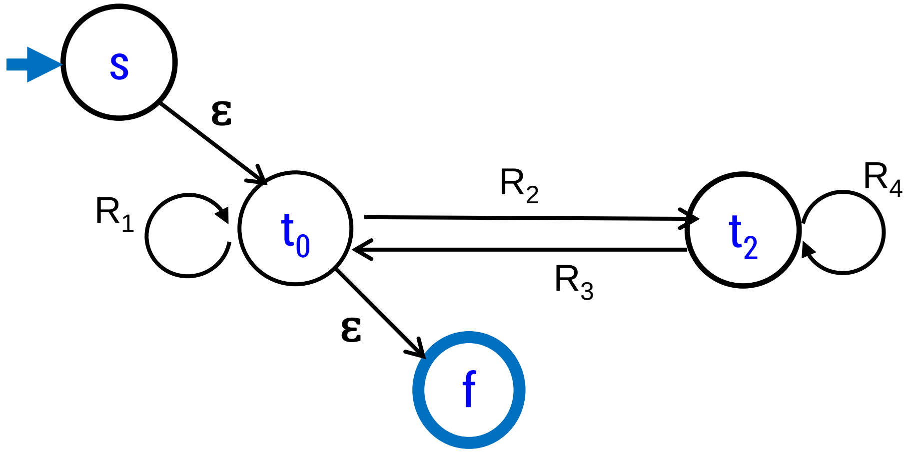
Label edges with regular expressions

$t_0 \rightarrow t_1 \rightarrow t_0$: 10*2
$t_0 \rightarrow t_1 \rightarrow t_2$: 10*1
$t_2 \rightarrow t_1 \rightarrow t_0$: 20*2
$t_2 \rightarrow t_1 \rightarrow t_2$: 20*1

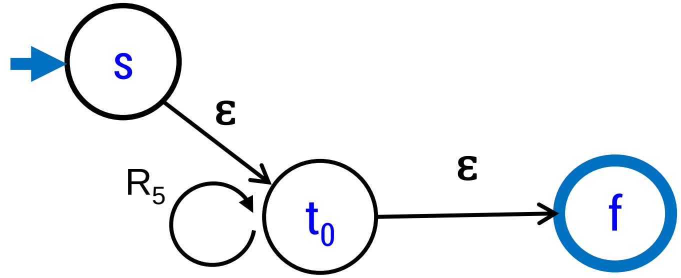


finite automaton without t_1

R_1: $0 \cup 10^*2$
R_2: $2 \cup 10^*1$
R_3: $1 \cup 20^*2$
R_4: $0 \cup 20^*1$



R_5: $R_1 \cup R_2R_4{}^*R_3$



Final regular expression:
$(0 \cup 10^*2 \cup (2 \cup 10^*1)(0 \cup 20^*1)^*(1 \cup 20^*2))^*$

irregular language!

B = {binary palindromes} can't be recognized by any DFA

Why is this language not regular?

Intuition (**not a proof**):

Q: What would a DFA need to keep track of to decide the language?

A: It would need to keep track of the "first part" of the input in order to check the second part against it… but there are an infinite # of possible first parts and we only have finitely many states.

How do we prove it?

B = {binary palindromes} can't be recognized by any DFA

Consider the infinite set of strings

$$S = \{1, 01, 001, 0001, 00001, \dots\} = \{0^n1 : n \geq 0\}$$

That's a nice set of first parts to have to remember but how can we argue that a DFA does the wrong thing for B?

- Show that some **x** ∈ B and some y ∉ B both must end up at the *same* state of the DFA

That state can't be

- a final state since then y is accepted: error on y
- a non-final state since then x is rejected: error on x

B = {binary palindromes} can't be recognized by any DFA

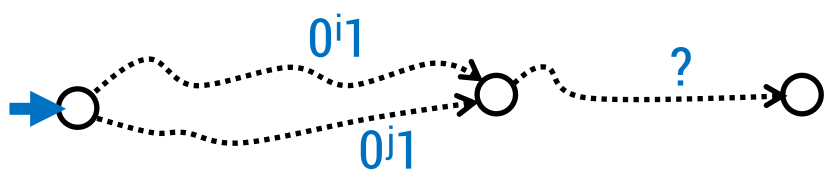
Consider the infinite set of strings

$$S = \{1, 01, 001, 0001, 00001, \dots\} = \{0^n 1 : n \geq 0\}$$

Suppose we are given an arbitrary DFA **M**.

- Goal: Show that some **x** ∈ **B** and some **y** ∉ **B** both must end up at the *same* state of **M**

Since **S** is infinite we know that two different strings in **S** must land in the same state of **M**, call them $\mathbf{0^i1}$ and $\mathbf{0^j1}$ for $i \neq j$.



- That also must be true for $\mathbf{0^i1z}$ and $\mathbf{0^j1z}$ for any $\mathbf{z} \in \{\mathbf{0,1}\}^*$ **!**

In particular, with $\mathbf{z=0^i}$ we get that $\mathbf{0^i10^i}$ and $\mathbf{0^j10^i}$ end up at the same state of **M**. Since $\mathbf{0^i10^i} \in \mathbf{B}$ and $\mathbf{0^j10^i} \notin \mathbf{B}$ (because $i \neq j$), M does not recognize **B**. ∴ no DFA can recognize **B**.

showing a language L is not regular

1. Find an infinite set **S**=$\{s_0,s_1,\ldots,s_n,\ldots\}$ of string prefixes that you think will need to be remembered separately

2. "Let **M** be an arbitrary DFA. Since **S** is infinite and **M** is finite state there must be two strings s_i and s_j in **S** for some $i \neq j$ that end up at the same state of **M**."

   Note: You don't get to choose which two strings s_i and s_j

3. Find a string **t** (typically depending on s_i and/or s_j) such that

   s_it is in **L**, and s_jt is not in **L** or s_it is not in **L**, and s_jt is in **L**

4. "Since s_i and s_j both end up at the same state of **M**, and we appended the same string **t**, both s_it and s_jt end at the same state of **M.** Since $s_it \in$ **L** and $s_jt \notin$ **L**, **M** does not recognize **L**."

5. "Since **M** was arbitrary, no DFA recognizes **L**."

$A = \{0^n 1^n : n \geq 0\}$ cannot be recognized by any DFA

1. Find an infinite set **S**=$\{\mathbf{s_0},\mathbf{s_1},\ldots,\mathbf{s_n},\ldots\}$ of string prefixes that you think will need to be remembered separately

2. "Let **M** be an arbitrary DFA. Since **S** is infinite and **M** is finite state there must be two strings $\mathbf{s_i}$ and $\mathbf{s_j}$ in **S** for some $\mathbf{i} \neq \mathbf{j}$ that end up at the same state of **M**."

3. Find a string **t** (typically depending on $\mathbf{s_i}$ and/or $\mathbf{s_j}$) such that

   $\mathbf{s_i t}$ is in **L**, and

   $\mathbf{s_j t}$ is not in **L**

4. "Since $\mathbf{s_i}$ and $\mathbf{s_j}$ both end up at the same state of **M**, and we appended the same string **t**, both $\mathbf{s_i t}$ and $\mathbf{s_j t}$ end at the same state of **M.** Since $\mathbf{s_i t} \in \mathbf{L}$ and $\mathbf{s_j t} \notin \mathbf{L}$, **M** does not recognize **L**."

5. "Since **M** was arbitrary, no DFA recognizes **L**."

$A = \{0^n 1^n : n \geq 0\}$ cannot be recognized by any DFA

1. Find an infinite set **S**=$\{$**s**$_0$,**s**$_1$,…,**s**$_n$,…$\}$ of string prefixes that you think will need to be remembered separately

$$S = \{0^n : n \geq 0\}$$

2. "Let **M** be an arbitrary DFA. Since **S** is infinite and **M** is finite state there must be two strings **s**$_i$ and **s**$_j$ in **S** for some **i** ≠ **j** that end up at the same state of **M**."

$$s_i = 0^i, s_j = 0^j$$

3. Find a string **t** (typically depending on **s**$_i$ and/or **s**$_j$) such that

   s$_i$**t** is in **L**, and

   s$_j$**t** is not in **L**

$$t = 1^i$$

4. "Since **s**$_i$ and **s**$_j$ both end up at the same state of **M**, and we appended the same string **t**, both **s**$_i$**t** and **s**$_j$**t** end at the same state of **M**. Since **s**$_i$**t** ∈ **L** and **s**$_j$**t** ∉ **L**, **M** does not recognize **L**."

5. "Since **M** was arbitrary, no DFA recognizes **L**."

another irregular language

$L = \{x \in \{0, 1, 2\}^*: x$ has an equal number of substrings 01 and 10$\}$.

Intuition: Need to remember difference in # of **01** or **10** substrings seen, but only hard to do if these are separated by **2**'s.

1. Let **S**=$\{\varepsilon$, 012, 012012, 012012012, ...$\}$ = $\{(012)^n : n \in \mathbb{N}\}$

2. Let **M** be an arbitrary DFA. Since **S** is infinite and **M** is finite, there must be two strings $(012)^i$ and $(012)^j$ for some $i \neq j$ that end up at the same state of **M**.

3. Consider appending string **t** = $(102)^i$ **to each of these strings.**

   Then $(012)^i (102)^i \in$ **L** but $(012)^j (102)^i \notin$ **L** since $i \neq j$

4. So $(012)^i (102)^i$ and $(012)^j (102)^i$ end up at the same state of **M** since $(012)^i$ and $(012)^j$ do. Since $(012)^i (102)^i \in$ **L** and $(012)^j (102)^i \notin$ **L**, **M** does not recognize **L**.

5. Since **M** was arbitrary, no DFA recognizes **L**.