Let's see whether we have languages we cannot recognize by an NFA/DFA.

consider the following languages.

$A = \{w : w \text{ has more } 0\text{'s than } 1\text{'s}\}$

$B = \{a^n b^n \mid n \geq 0\}$

$C = \{w \mid w \text{ has equal number of } 0\text{'s and } 1\text{'s}\}$

When you consider language A,
If you want to build a DFA
that recognizes A, the machine
needs to keep track of # of
0's or 1's, and depending on the
string this value could be different.
Therefore, a machine with limited
amount of memory cannot do it.

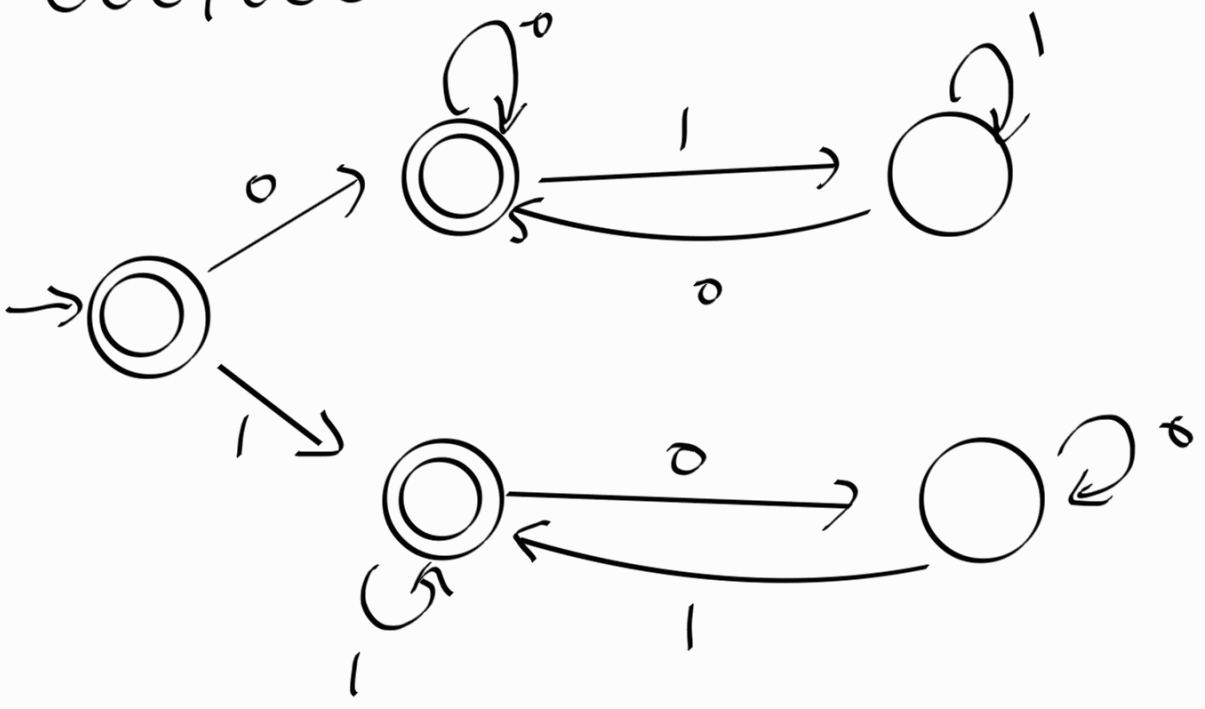
However, just because a language appears to require unbounded memory does not mean that the language is not regular.

$C = \{ w \mid w$ has an equal # of 0's and 1's $\}$

$D = \{ w \mid w$ has equal # of occurrences of 01 and 10 $\}$

ε ✓ 0 ✓ 00000 ✓ 1 ✓ 111111 ✓

0001000



Pumping Lemma

- Basically, this theorem states that regular languages have a special property.

- If we can show that a language does not have this property, then the language is not regular.

The basic idea of this property is that all strings in a regular language can be "pumped", If they are at least as long as a certain special value, called the pumping length.

Pumping Lemma

If A is a regular language, then there is a number p (the pumping length) where if s is any string in A of length at least p, then s may be divided into three pieces, $s = xyz$, satisfying the following conditions.

1. for each $i \geq 0$, $xy^iz \in A$
2. $|y| > 0$, and
3. $|xy| \leq p$

- $|s|$ represents the length of the string
- y^i means i copies of string y.

A is regular $\Rightarrow$ [$\exists p \in \mathbb{N}$: $s \in A \wedge |s| \geq p \Rightarrow s = xyz$]

1.
2.
3.

Proof idea:

Assume A is regular.
Then A can be recognized by some DFA, let's call it M.

We pick p to be the # of states in DFA M (that recognize language A).

Then pick any string s in A of length at least p.

* If no such string exist, then the claim is vacuously true.

Since the size of s is larger than or equal to p, then in its execution path to an accepting state, we must see a repeated state.

$S = xyz$

xz

$xyyz$

$xy^i z \in A$

since this path leads to an accepting state, this path should accept strings, $xz, xyz, xy^2z, xy^3z, \ldots, xy^iz, \ldots$

Question! Can we use pumping lemma to show that a particular language is regular?

You do not use pumping lemma to show a language is regular.

In fact, we cannot do this as this lemma is not bidirectional.

Just because a language follows pumping lemma it does not mean it is regular.

non-regular but follows pumping lemma.

Ex: $L = \{ab^n c^n : n \geq 0\} \cup \{a^k w : k \neq 1 \text{ and } w \in \Sigma^* \text{ does not start with a}\}$

$\Sigma = \{a, b, c\}$

But we can use this to show a language is not regular.

Applications of pumping lemma.

Pumping Lemma is used to show that a particular language is not regular.

Following is the procedure that we can follow to show a particular language is not regular.

Suppose you are given the language A.

Language A

1. Language A's size n should be infinite, because if it was finite, then A should be regular.

2. Let's assume A is regular.

3. Then, there should be a DFA that recognizes language A. Let's call this machine M.

4. Since M is a DFA, # states in M should be finite.

5. Since A is an infinite language, there must be at least one string in A that is of length at least p.

6. let's pick some s, s.t $|s| \geq p$.

Existence of s in A implies that there is a path to accepting state that reuses at least one state in the machine.

then any string that uses this path must be accepted by this machine, e.g.,

$$xz, xyz, xy^2z, xy^3z, \ldots$$

The trick is to pick a string s from A, s.t, once it is decomposed into $s = xyz$, the strings that we generate by pumping y are no longer in language A.

* Remember there could be multiple ways to decompose s into xyz.

Therefore, we need to show that every decomposition of s leads to creating new string that are not in A.

If you create strings that are accepted by M but not in A, then we have a contradiction.

Example! prove $A = \{0^n 1^n \mid n \geq 0\}$ is not regular.

Assume A is regular.

Suppose pumping length of language A is p.

(Remember this is a fancy way of saying the DFA that recognize A has p states)

Now, the goal is to pick a string s in A of length at least p such that it violates the pumping lemma.

Note: I made a mistake saying that following is bidirectional.

A is regular $\Rightarrow$ A follows pumping Lemma

This is not bidirectional.

$L = \{a b^n c^n : n \geq 0\} \cup \{a^k w : k \neq 1$ and $w \in \Sigma^*$ does not start with a $\}$

This language follows pumping lemma but is not regular.