

# Pumping Lemma for Regular Languages

some languages are not regular!

Sipser pages 77 - 82

# Are all Languages Regular

We have seen many ways to specify Regular languages

Are all languages Regular languages?

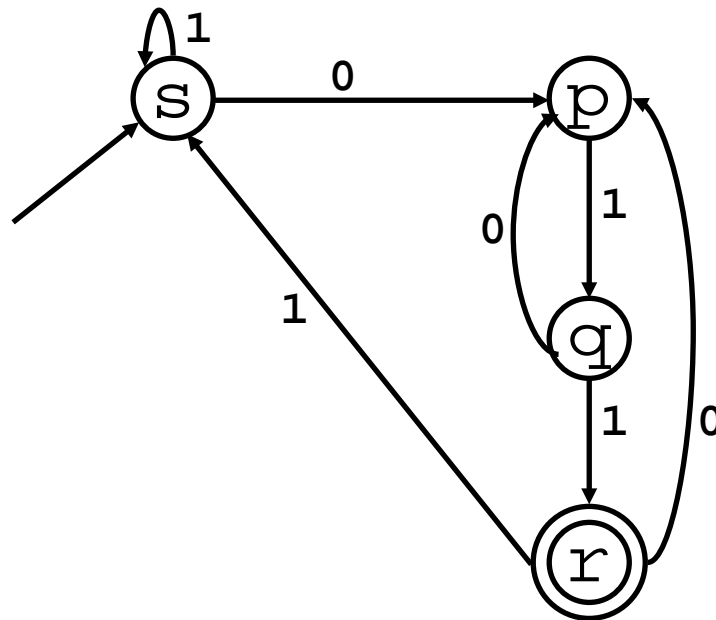
The answer is No,  
How can we tell?

A language is regular if we can describe it using any of the formalisms we have studied.

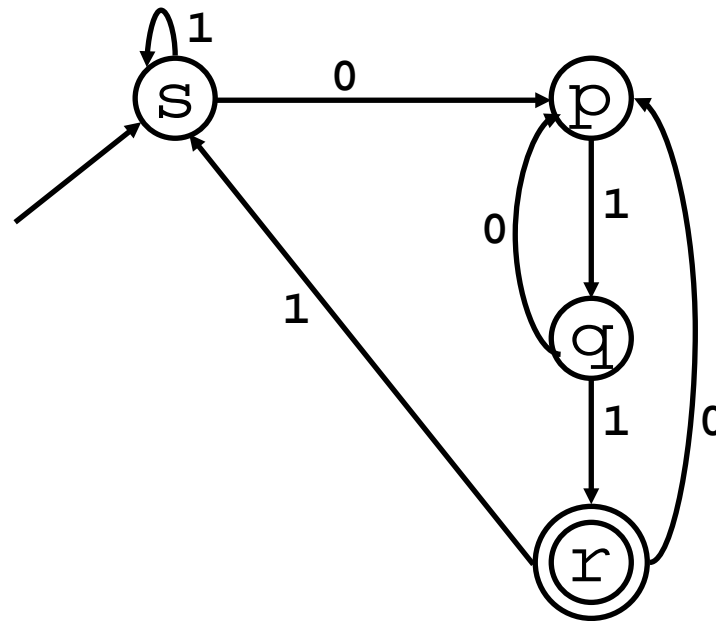
If we can't describe it, does that mean it is not regular? Maybe we're not clever enough.

# Importance of loops

Consider this DFA. The input string 01011 gets accepted after an execution that goes through the state sequence  $s \rightarrow p \rightarrow q \rightarrow p \rightarrow q \rightarrow r$ . This path contains a loop (corresponding to the path  $[p, q, p]$  and the substring 10) that starts and ends at  $p$ . There are two simple ways of modifying this path *without changing its beginning and ending states*:



- (1) delete the loop from the path;
- (2) instead of going around the loop once, do it several times. As a consequence, we see that all strings of the form  $0(10)^i11$  (where  $i \geq 0$ ) are accepted.



# Long paths must contain a loop

Suppose  $n$  is the number of states of a DFA.

Then every path of length  $n$  or more makes at least  $n+1$  visits to a state and therefore must visit some state twice.

Thus, *every path of length  $n$  or longer must contain a loop.*

# Strategy

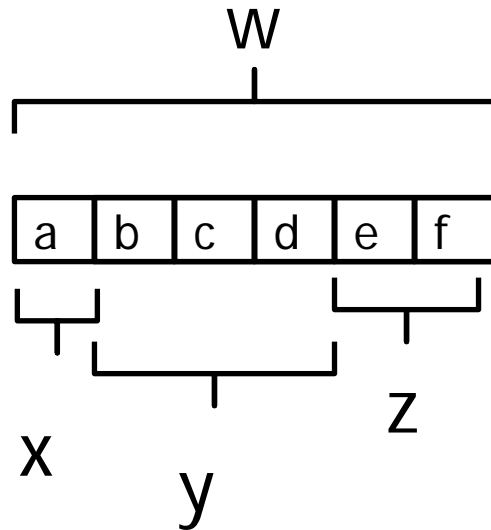
Every long string in a regular language must have a loop.

Regular Languages with loops exhibit certain kinds of patterns that are distinctly regular.

Languages with long strings that do not adhere to the loop patterns for regular languages cannot be regular.

# Pumps

Suppose  $\mathcal{L}$  is a regular language,  $w$  is a string in  $\mathcal{L}$ , and  $y$  is a non-empty substring of  $w$ . Thus,  $w = xyz$ , for some strings  $x, z$ . We say that  $y$  is a *pump* in  $w$  if all strings  $xy^iz$  (that is,  $xz, xyz, xyyz, xy^2yz, \dots$ ) belong to  $\mathcal{L}$ .



aef

ab**cd**ef

ab**cd****bc**def

ab**cd****bc****bc**def

# Pumping Lemma

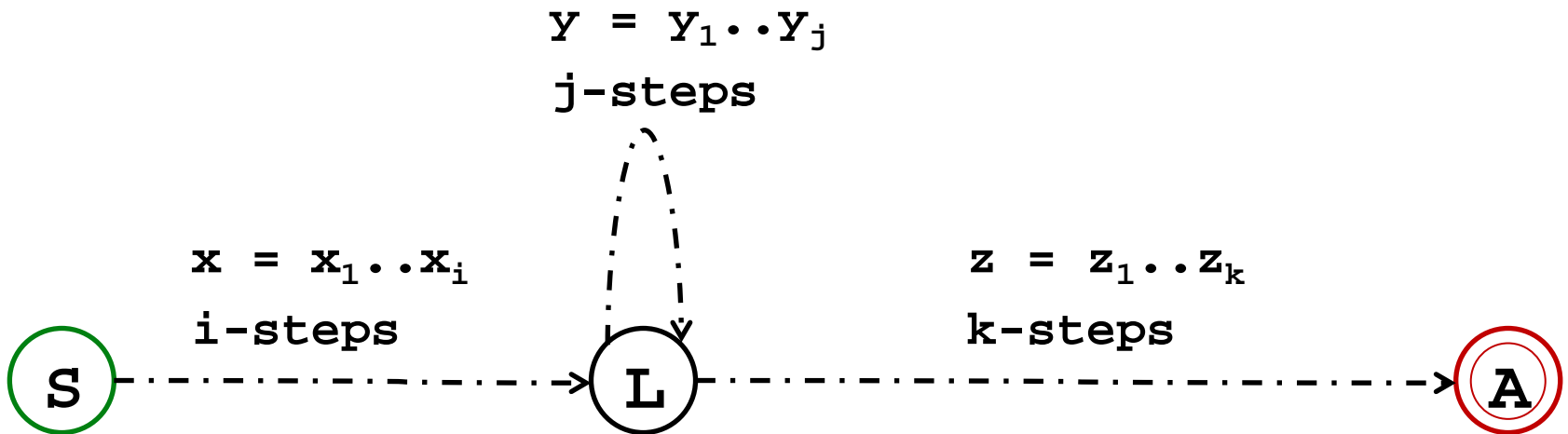
Let  $L$  be a regular language. Then there exists a number  $n$  such that all  $w \in L$  where  $|w| \geq n$ , there exists a prefix of  $w$  whose length is less than  $n$  which contains a pump. Formally: If  $w \in L$  and  $|w| \geq n$  then  $w = xyz$  such that

1.  $y \neq \varepsilon$
2.  $|xy| \leq n$  (xy is the prefix)
3.  $xy^iz \in L$

**Definition.** The number  $n$  associated to the regular language  $L$  as described in the Pumping Lemma is called the *pumping constant* of  $L$ .



# Proof strategy



Let the DFA,  $A$ , have  $M$  states, and let the string  $w$  accepted by  $A$  have length  $len$ , where  $len > M$ .

The path starts in  $S$  (the start state) and ends in  $A$  (an accepting state)

The accepting path must have a loop! Why?

Let the loop go through state  $L$ .

$$w = xyz \quad |w| = i+j+k \quad i+j+k > M$$

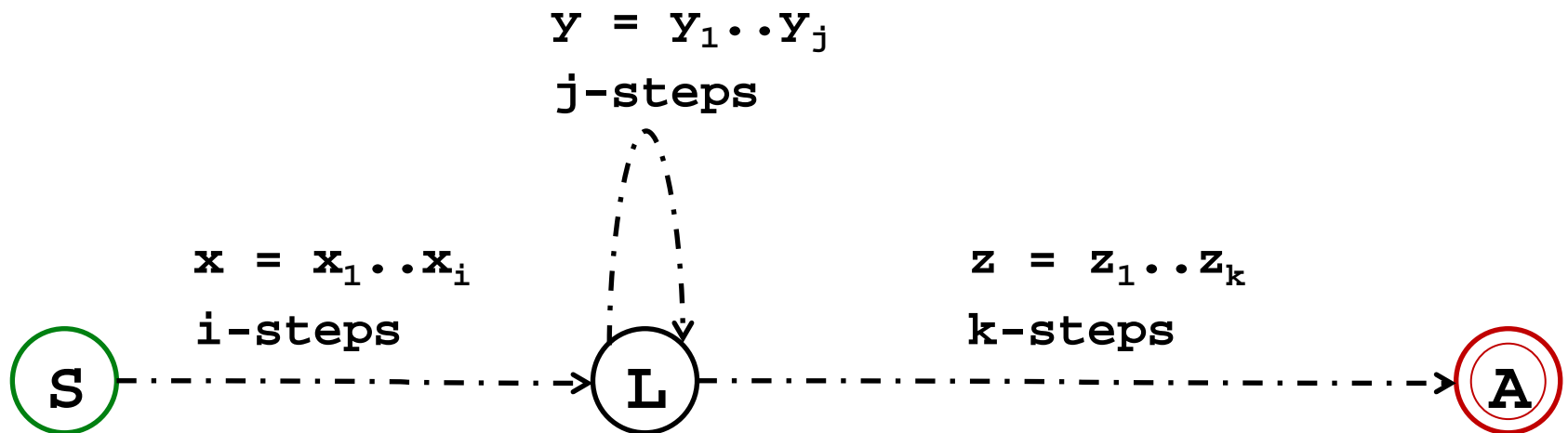
# Proof

$w \in L$ ,  $|w| \geq n$ ,  $w = xyz$  such that 1.  $y \neq \varepsilon$  2.  $|xy| \leq n$  3.  $xy^iz \in L$

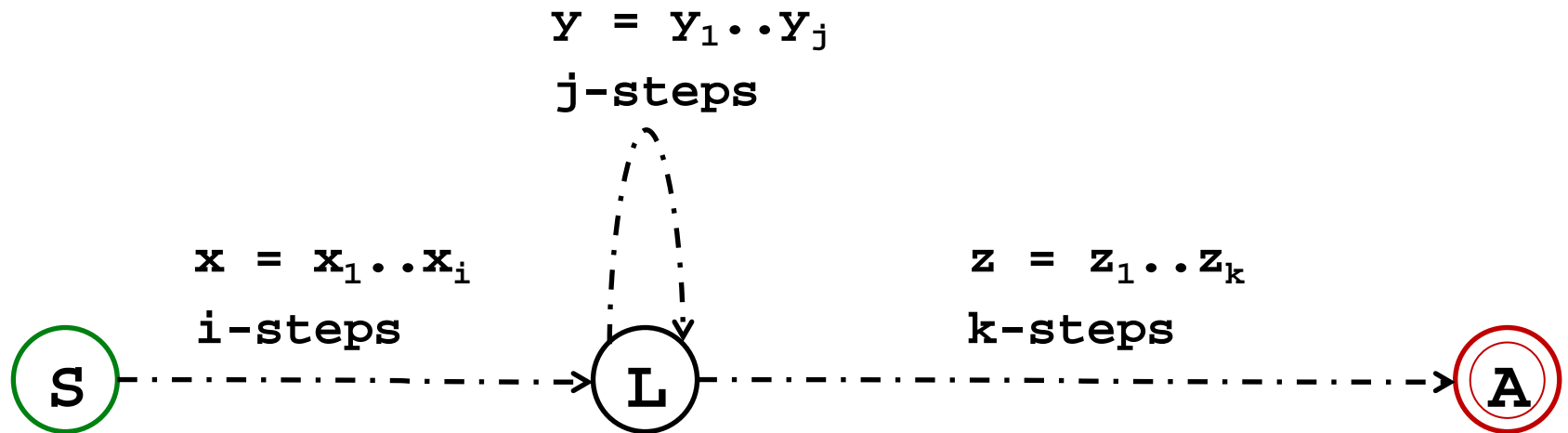
Let the DFA have  $m$  states. Let  $|w| \geq m$ .

Consider the path from the start state  $s$  to the (accepting) state  $\delta(s, w) = A$ .

Just following the first  $m$  arcs, we make  $m+1$  total visits to states, so there must be a loop formed by some of these arcs.



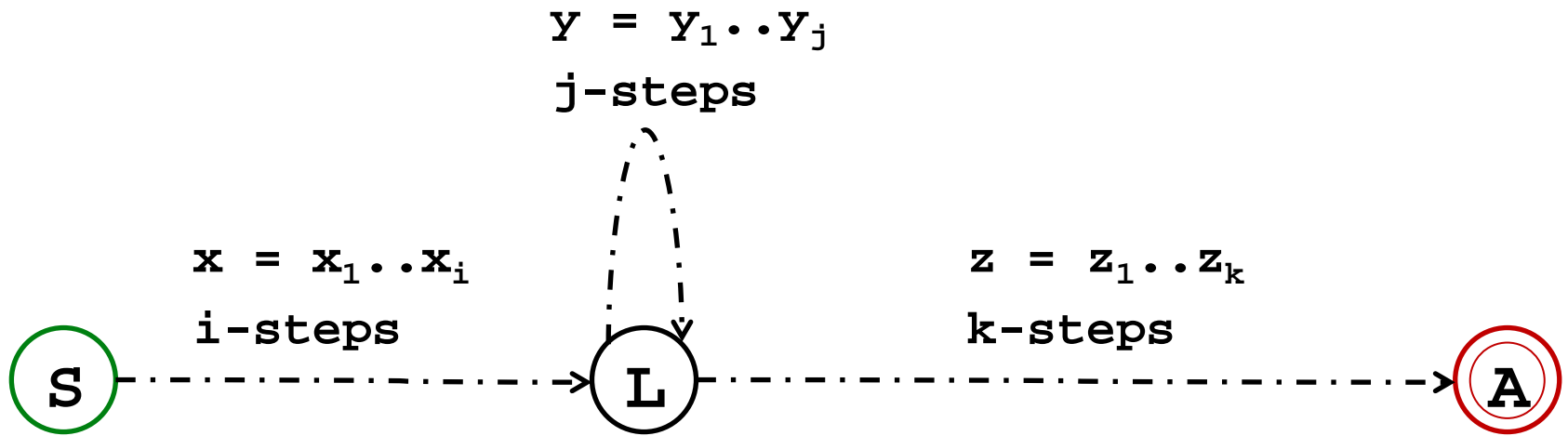
# For all DFA there exist a number N



Let  $n$  be number greater than  $i+j$   
Anything longer than  $i+j$  will also  
have a loop.

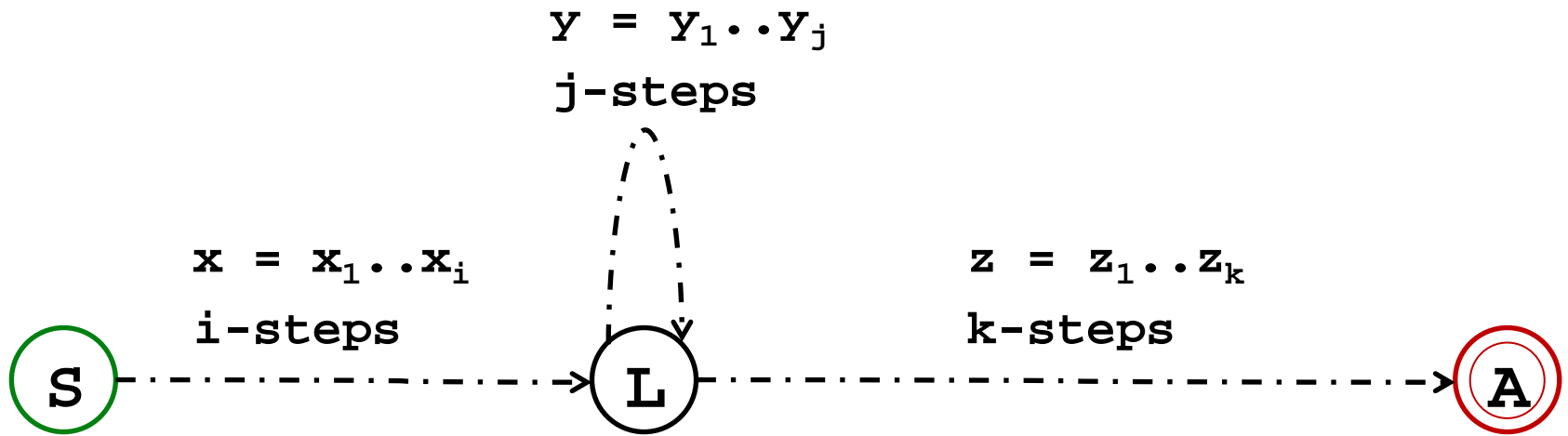
At least the  $L$  to  $L$  loop, but  
possibly more.

# 1. $y \neq \varepsilon$



If  $L$  really is a loop, then  $y$  has at least one symbol, so  $y \neq \varepsilon$

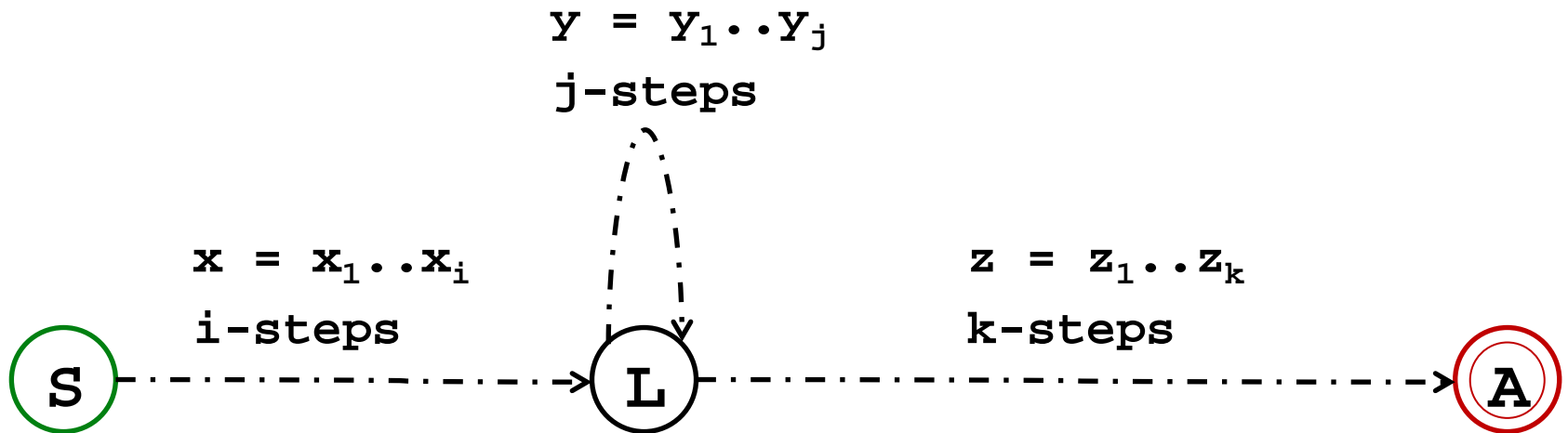
## 2. $|xy| \leq n$



We choose  $n$  to be greater than  $i+j$

So  $|xy| \leq n$

### 3. $xy^iz \in L$



$xy^iz \in L$

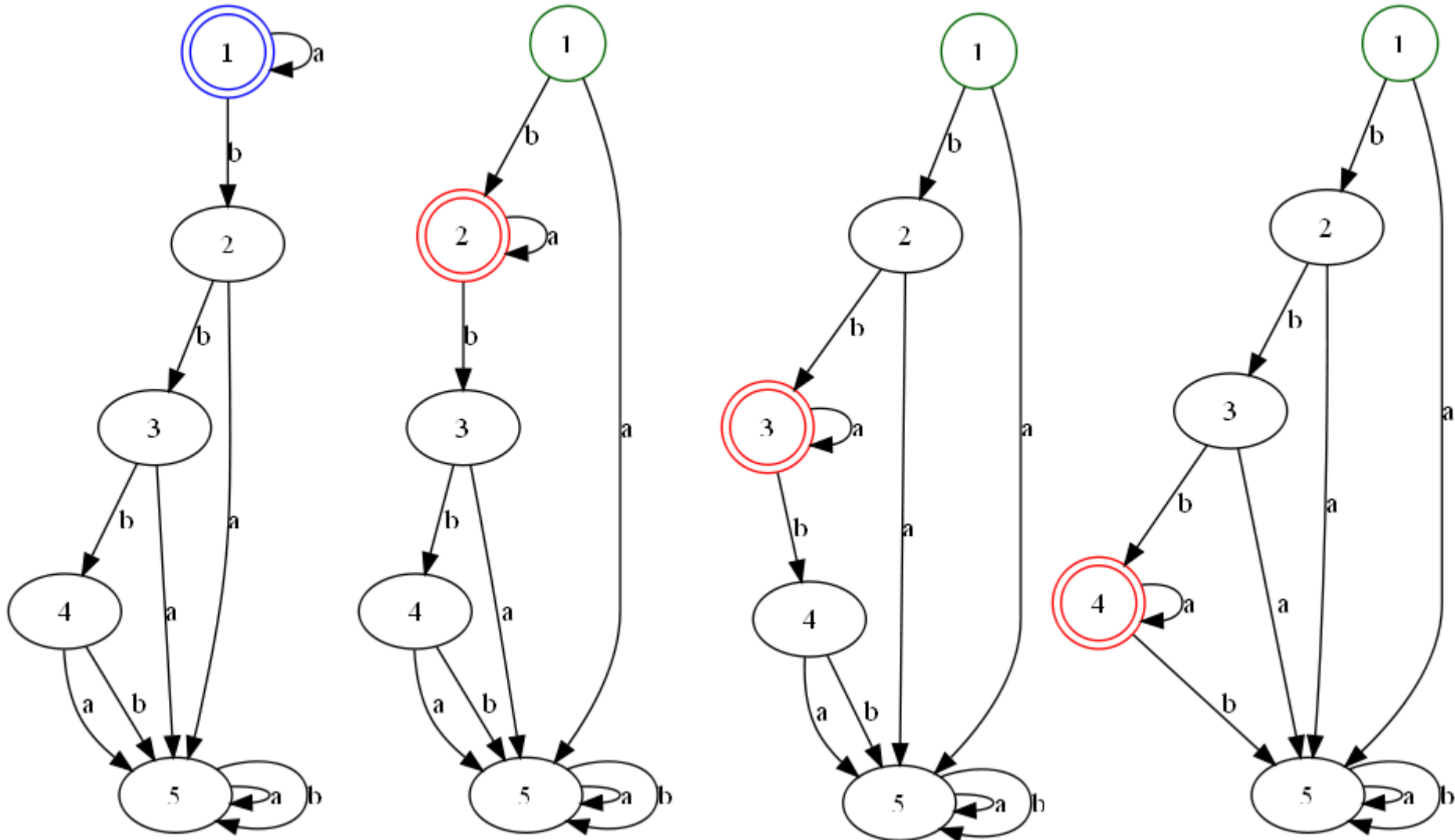
For  $i=0$  skip the loop

For  $i=n$  go around the loop  $n$ -times

If  $y$  is a loop,

1. It starts at some state  $L$  and  $\underline{\delta}(L,y) = L$  (that is why it's a loop!)
2. We know that  $\underline{\delta}(L,yz) = A$ .
3. Thus  $\underline{\delta}(S,x) = L$ , Thus for each  $i$   $\underline{\delta}(L,y^i) = L$ , and were done.

# Why the pumping constant can be smaller than the number of states



# Proving non-regularity

To prove that a given language,  $L$ , is not regular, we use the Pumping Lemma as follows.

1. We use a proof by contradiction.
2. We assume that  $L$  is regular.
3. It must be recognized by a DFA.
4. That DFA must have a pumping constant  $N$
5. We carefully choose a string longer than  $N$  (so the lemma holds)
6. Show that pumping that string leads to a contradiction
7. Thus our original assumption that  $L$  was regular must be false. So  $L$  is not regular!



# Strategy

Assume  $L$  is regular (we are arguing by contradiction!).

Let  $n$  be the pumping constant of  $L$ .

Making no other assumptions about  $n$  (we don't know what it is exactly)

We need to produce a string  $w \in L$  of length  $\geq n$  that does not contain a pump in its  $n$ -prefix.

This  $w$  depends on  $n$ ; we need to give  $w$  for any value of  $n$

The Key is to carefully crafting  $w$  as a function of  $n$ , so that it is clearly longer than  $n$

## Strategy 2

There are many substrings of the  $n$ -prefix of our chosen  $w$

We must demonstrate that *none of them is a pump*.

Typically, we do this by writing  $w = xuy$ , a decomposition of  $w$  into three substrings about which we can only assume that  $u \neq \varepsilon$  and  $|xu| \leq n$ .

Then we must show that *for some concrete  $i$*  (zero or greater) the string  $xu^iy$  does not belong to  $L$ .

## Skill required

Notice the game-like structure of the proof. Somebody gives us  $n$ . Then we give  $w$  of length  $\geq n$ . Then our opponent gives us a non-empty substring  $u$  of the  $n$ -prefix of  $w$  (and with it the factorization  $w = xuy$  of  $w$ ). Finally, we choose  $i$  such that  $xu^i y \notin L$ .

Our first move often requires ingenuity: We must find  $w$  so that we can successfully respond to whatever our opponent plays next.

# Example 1

We show that  $L = \{0^k 1^k \mid k=0,1,2, \dots\}$  is not regular.

Assuming the Pumping Lemma constant of  $L$  is  $n$ , we take  $w = 0^n 1^n$ . We need to show that there are no pumps in the  $n$ -prefix of  $w$ , which is  $0^n$ . If  $u$  is a pump contained in  $0^n$  then  $0^n = xuz$ , and  $xuuz$  must also be in the language. But since  $|u| > 0$ , if  $|xuz| = n$  then  $|xuuz| = m$  where  $m > n$ . So we obtain a string  $0^m 1^n$  with  $m > n$ , which is obviously not in  $L$ , so a contradiction is obtained, and our assumption that  $0^k 1^k$  is regular must be false.

*Note.* The same choice of  $w$  and  $i$  works to show that the language:

$L = \{w \in \{0,1\}^* \mid w \text{ contains equal number of 0s and 1s}\}$   
is not regular either.

## Example 2

We show that  $L = \{ uu \mid u \in \{a,b\}^* \}$  is not regular. Let  $n$  be the pumping constant. Then we choose  $w = a^n b a^n b$  which clearly has length greater than  $n$ .

The initial string  $a^n$  must contain the pump,  $u$ . So  $w = xuyba^n b$ , and  $xuyb = a^n b$ . But pumping  $u$  0 times it must be the case that  $xyba^n b$  is in  $L$  too. But since  $u$  is not  $\varepsilon$ , we see that  $xyb \neq a^n b$ , since it must have fewer  $a$ 's. Which leads to a contradiction. Thus our original assumption that  $L$  was regular must be false.

*Question.* If in response to the given  $n$  we play  $w = a^n a^n$ , the opponent has a chance to win. How?