Theorem. *For every $L \subseteq \Sigma^*$, if there is an infinite set S of pairwise L-distinguishable strings, then L cannot be accepted by FA.*

Proof sketch. If S is infinite then for every n, S has a subset with n elements. If M was FA accepting L, then previous theorem would say that for every n, M would have at least n states. No FA has this property. $\square$

Example. *For every pair of distinct x and y in $\{a, b\}^*$, x, and y are distinguishable wrt PAL.*

Consider the language of palindromes, PAL, over $\{a, b\}^*$. Let's take two strings x, and y.

- *If $x \neq y$, $|x| = |y|$, then $r(x)$ distinguishes between x, and y, because $xr(x) \in PAL$, and $yr(x) \notin PAL$.*

- *If $|x| \neq |y|$, w.l.o.g. we assume $|x| < |y|$. If x is not a prefix of y, then $xr(x) \in PAL$, and $yr(x) \notin PAL$.*

- *If x is a prefix of y, then $y = xz$ for some z. If we choose symbol σ so that $z\sigma$ is not a palindrome then*

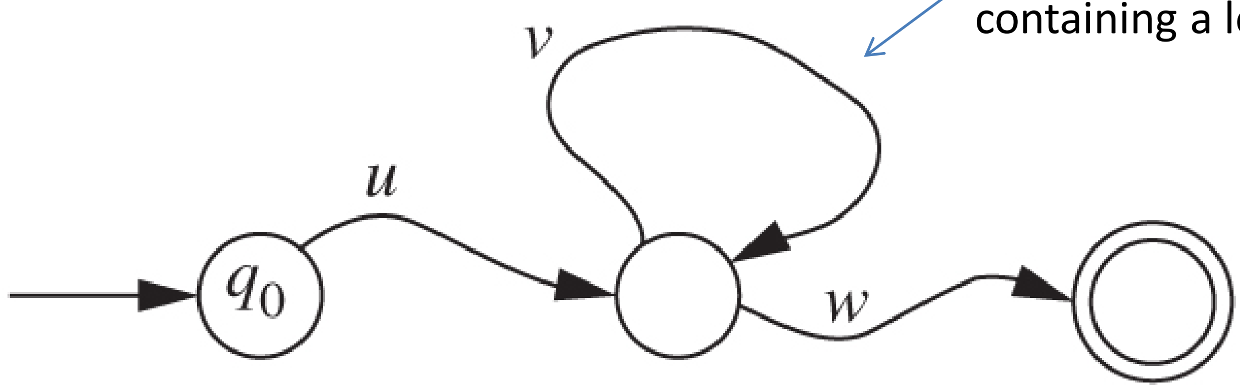
$$x\sigma r(x) \in PAL \quad \text{and} \quad y\sigma r(x) = xz\sigma r(x) \notin PAL,$$

PAL cannot be accepted by FA because we can find infinitely many PAL-distinguishable strings. (Requires infinite memory to remember $r(\cdot)$)

The Pumping Lemma

- Suppose that $M=(Q, \Sigma, q_0, A, \delta)$ is an FA accepting L and that **it has *n* states**
  - If it **accepts** a string x such that $|x| \geq n$, then by the time n symbols have been read, M must have entered some state more than once; i.e., there must be two different prefixes u and uv such that $\delta^*(q_0,u)= \delta^*(q_0,uv)$

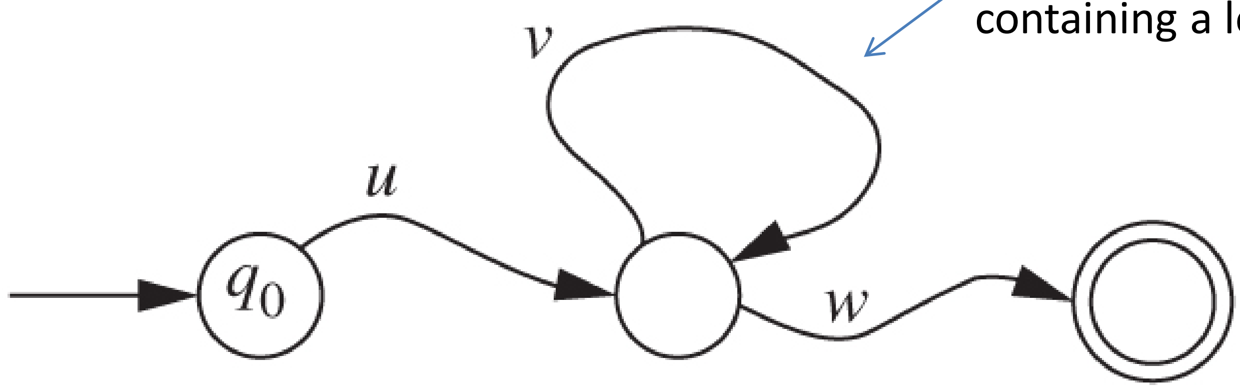
There must be a path containing a loop



The Pumping Lemma (cont'd.)

- This implies that there are many more strings in L, because we can traverse the loop v any number of times (including leaving it out altogether)
- In other words, all of the strings uv^iw for $i \geq 0$ are in L
- This fact is known as the Pumping Lemma for Finite Automata (or for Regular Languages)

There must be a path containing a loop



The Pumping Lemma

- **Theorem**: Suppose L is a language over Σ

If L is accepted by the FA $M=(Q, \Sigma, q_0, A, \delta)$, and $|Q| = n$, then for every x in L satisfying $|x| \geq n$, there are three strings u, v, and w such that $x = uvw$ and

  - $|uv| \leq n$
  - $|v| > 0$ (i.e. $v \neq \Lambda$)
  - For every $i \geq 0$, the string uv^iw belongs to L

- The way we found n was to take the number of states in an FA accepting L. In many applications we don't need to know this, only that there is such an n

The Pumping Lemma (cont'd.)

- The most common application of the pumping lemma is to show that a language *cannot* be accepted by an FA, because it doesn't have the properties that the pumping lemma says are required for every language that can be.
- The proof is by contradiction. We suppose that the language can be accepted by an FA, and we let $n=|Q|$ be the integer in the pumping lemma
- Then we choose a string x with $|x| \geq n$ to which we can apply the lemma so as to get a contradiction

Example: language that cannot be accepted by an FA, one way to use the pumping lemma

- Let L be the language $AnBn = \{a^i b^i \mid i \geq 0\}$; let us prove that it cannot be accepted by an FA
  - Suppose, for the sake of contradiction, that L is accepted by an FA; let n be as in the pumping lemma
  - Choose $x = a^n b^n$; then $x \in L$ and $|x| \geq n$
  - Therefore, by the pumping lemma, there are strings u, v, and w such that $x = uvw$ and the 3 conditions hold
  - Because $|uv| \leq n$ and x starts with n a's, all the symbols in u and v are a's; therefore, $v = a^k$ for some $k > 0$
  - $uvvw \in L$, so $a^{n+k} b^n \in L$. This is our contradiction, and we conclude that L cannot be accepted by an FA

Example 2: language that cannot be accepted by an FA, one way to use the pumping lemma

- Let's show $L = \{a^{i^2} \mid i \geq 0\}$ is not accepted by an FA
  - Suppose L is accepted by an FA, and let n be the integer in the pumping lemma
  - Choose $x = a^{n^2}$
  - $x = uvw$, where $0 < |v| \leq n$
  - Then $n^2 = |uvw| < |uv^2w| = n^2 + |v| \leq n^2 + n < (n+1)^2$
  - This is a contradiction, because $|uv^2w|$ must be i^2 for some integer i (because $uv^2w \in L$), but there is no integer i whose square is strictly between n^2 and $(n+1)^2$

The Pumping Lemma (cont'd.)

- There are other languages that are not accepted by any FA, among them:
  - *Balanced*, the set of balanced strings of parentheses
  - *Expr*, the language of simple algebraic expressions
  - The set of legal C programs
- In all three examples, because of the nature of these languages, a proof using the pumping lemma might look a lot like the proof for *AnBn*, our first example
- For example, this string "main(){{{ … }}}" cannot be accepted by an FA (because of $\{^n\}^n$).

The Pumping Lemma (cont'd.)

- We can formulate several "decision problems" involving the language L accepted by an FA
  - The membership problem (Given x, is $x \in L(M)$?)
  - Given an n-state FA M, is the language $L(M)$ empty?
    - It follows from the PL that this can be solved by looking at all possible strings of length 0 to n -1; if none of those is accepted, the language is empty, i.e., if we find x that is longer than n, we can always extract a middle part v.
  - Given an n-state FA M, is $L(M)$ infinite?
    - The pumping lemma implies that the language is infinite if and only if at least one of the strings with length from n to $2n$ -1 is accepted

Example: Given two FAs M_1 and M_2, are there any strings that are accepted by neither?

- – We know how to construct an FA for the complement of a language L for a given FA, i.e., for $\overline{L} = \Sigma^* - L$.
  – Apply this for finding $\overline{M}_1$, and $\overline{M}_2$, accepting $\overline{L(M_1)}$, and $\overline{L_(M_2)}$.
  – We can construct an FA accepting $\overline{L(M_1)} \cap \overline{L_(M_2)}$.
  – Run the algorithm for determining if this FA accepts any strings.