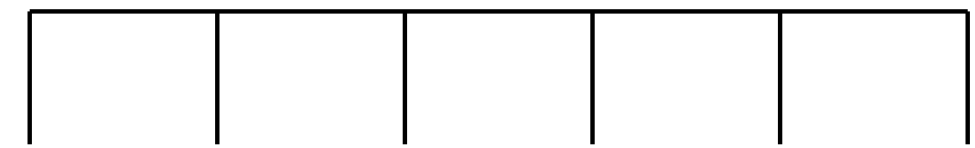
Pigeonhole principle

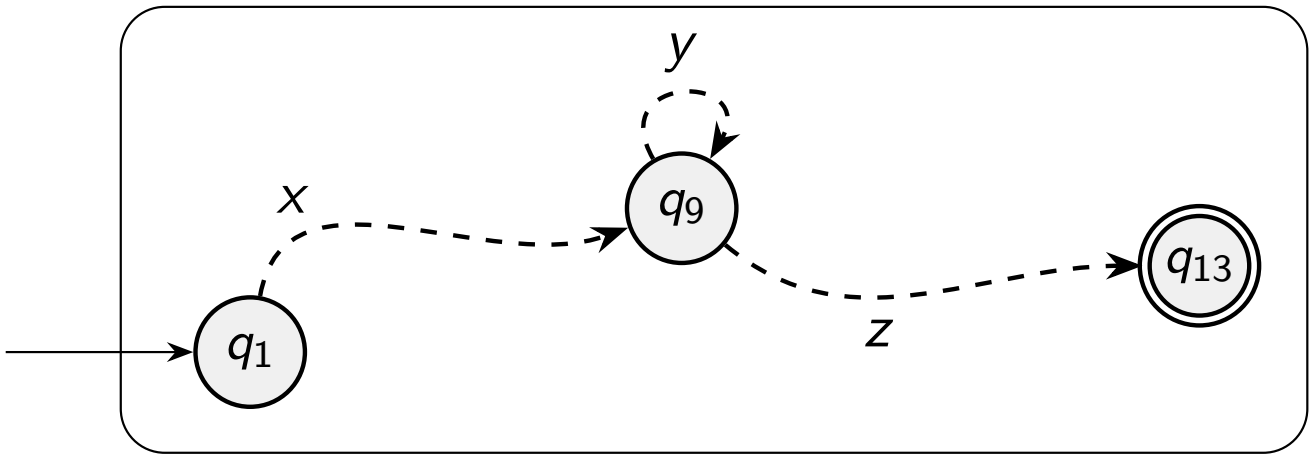
- Consider language L, which contains word $w \in L$.
- Consider an FA which accepts L, with $n < |w|$ states.
- Then, when accepting w, the FA must visit at least one state twice.
- This is according to the pigeonhole (a.k.a. Dirichlet) principle:
  - If $m > n$ pigeons are put into n pigeonholes, there's a hole with more than one pigeon.
  - That's a pretty fancy name for a boring observation...





Languages with unbounded strings

- Consequently, regular languages with unbounded strings can only be recognized by FA (finite! bounded!) automata if these long strings loop.



- The FA can enter the loop once, twice, . . . , and not at all.
- That is, language L contains **all** $\{xz, xyz, xy^2z, xy^3z, \ldots\}$.

Pumping Lemma

- Theorem: Given a regular language L, there is a number p (called the **pumping number**) such that any string in L of length $\geq p$ is pumpable within its first p letters.
- **Intuition:** each such string contains a substring that can be repeated any number of times, with the resulting string still belonging to the language.
- In other words, for all $u \in L$ with $|u| \geq p$ we can write:
  - $u = xyz$ (x is a prefix, z is a suffix)
  - $|y| \geq 1$ (mid-portion y is non-empty)
  - $|xy| \leq p$ (pumping occurs in first p letters)
  - $xy^i z \in L$ for all $i \geq 0$ (can pump y-portion)
- If, on the other hand, there is no such p, then the language is not regular.

Pumping Lemma Example

- Let L be the language $\{0^n 1^n \mid n \geq 0\}$
- Assume (for the sake of contradiction) that L is regular
- Let **p** be the pumping length. **Let u be the string $0^p 1^p$.**
- Let's check string u against the pumping lemma:
- "In other words, for all $u \in L$ with $|u| \geq p$ we can write:
  - $u = xyz$ (x is a prefix, z is a suffix)
  - $|y| \geq 1$ (mid-portion y is non-empty)
  - $|xy| \leq p \rightarrow y = 0^+$ (pumping occurs in first p letters)
  - $xy^i z \in L$ for all $i \geq 0$ (can pump y-portion)"

$\rightarrow$ Then, xz or xyyz is not in L. Contradiction!

Let's make the example a bit harder...

- Let L be the language
  $\{w \mid w$ has an equal number of 0s and 1s$\}$
- Assume (for the sake of contradiction) that L is regular
- Let p be the pumping length. Let u be the string 0^p1^p.
- Let's check string u against the pumping lemma:
- "In other words, for all $u \in L$ with $|u| \geq p$ we can write:
  - $u = xyz$ (x is a prefix, z is a suffix)
  - $|y| \geq 1$ (mid-portion y is non-empty)
  - $|xy| \leq p$ (pumping occurs in first p letters)
  - $xy^iz \in L$ for all $i \geq 0$ (can pump y-portion)"

Harder example continued

- Again, $|xy| \leq p$ forces y to consist of **0s only**. Pump it there: xz and $xyyz$ no longer have as many 0s as 1s, so they are not in L.
- **What is harder?** In the previous example *every* split of u failed, so $|xy| \leq p$ was just a shortcut. Here it is *essential*: with $x = 0^{p-1}$, $y = 01$, $z = 1^{p-1}$, every xy^iz stays in L.
- The choice of u matters too: $u = (01)^p$ can be pumped forever (take $y = 01$) and yields no contradiction.
- There's another alternative proof for this example:
  - 0^*1^* is regular.
  - $\cap$ is a regular operation.
  - If L regular, then $L \cap 0^*1^*$ is also regular.
  - However, $L \cap 0^*1^*$ is the language we studied in the previous example (0^n1^n). A contradiction.

Now you try...

- Is $L_1 = \{ww \mid w \in (0 \cup 1)^*\}$ regular?
- Is $L_2 = \{1^n \mid n \text{ being a prime number}\}$ regular?