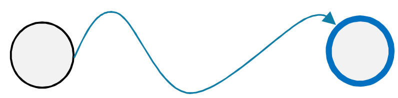
Forcing a Mistake

How do we know x, y must be in different states?

Well if one would be accepted and the other rejected, that would be a clear sign.

Or if there's some string z where xz is accepted but yz is rejected (or vice versa).

The machine is deterministic! If x and y take you to the same state, then xz and yz are also in the same state!



Full outline

1. Suppose for the sake of contradiction that L is regular. Then there is some DFA M that recognizes L.

2. Let S be [fill in with an infinite set of prefixes].

3. Because the DFA is finite and S is infinite, there are two (different) strings x, y in S such that x and y go to the same state when read by M *[you don't get to control x, y other than having them not equal and in S]*

4. Consider the string z [argue exactly one of xz, yz will be in L]

5. Since x, y both end up in the same state, and we appended the same z, both xz and yz end up in the same state of M. Since $xz \in L$and $yz \notin L$, M does not recognize L. But that's a contradiction!

6. So L must be an irregular language.

Outline for (*

1. Suppose for the sake of contradiction that L is regular. Then there is some DFA M that recognizes L.

2. Let S be (*

3. Because the DFA is finite and S is infinite, there are two (different) strings x, y in S such that x and y go to the same state when read by M Observe that $x = ($^{a} for some integer a, $y = ($$^{\text{b}}$ for some integer b with $a \neq b$.

4. Consider the string z [argue exactly one of xz, yz will be in L]

5. Since x, y both end up in the same state, and we appended the same z, both xz and yz end up in the same state of M. Since $xz \in L$and $yz \notin L$, M does not recognize L. But that's a contradiction!

6. So L must be an irregular language.

One more, just the key steps

What about $\{a^k b^k c^k : k \geq 0\}$?