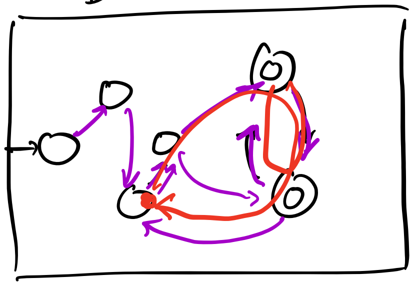
Sipser 77-79

D. $L(D)$ is regular and infinite.



- Pick $w \in L(D)$, $|w| \geq |Q|$.
- Computation of D on w

PL. (informally).

If D_A is the DFA for some infinite regular language A, then sufficiently long strings $w \in A$ make loops. (moreover, we can repeat the loop to get more strings in the language.)

Pumping Lemma (Formal).

Let A be a regular language. Then there is some number p such that all strings $s \in A$, with $|s| \geq p$ can be divided into 3 parts:

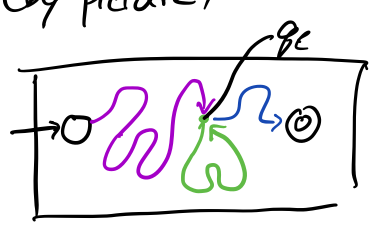
$$s = xyz$$

before loop, the loop, after loop.,

satisfying

1. $xy^iz \in A$ for all $i \geq 0$ // can go around loop as much as we want.
2. $|y| > 0$ // nontrivial loop
3. $|xy| < p$. // we find a loop before/as we touch $|Q|+1$ states.

Proof (by picture)



Consider a DFA D w/ $|Q|$ states

($L(D)$ finite? trivial/vacuous.)

Set $p = |Q|$

Pick $s \in L$ with $|s| \geq p = |Q|$. see path above.

Consider our path. Because $|s| \geq |Q|$, we repeat a state somewhere.

Let q_c be the first repeated state.

x – part of s before q_c

y – the loop (characters from q_c to q_c)

z – part of s after loop.

1. $xy^iz \in L$ for $i \geq 0$.
2. $|y| > 0$.
3. $|xy| < p = |$