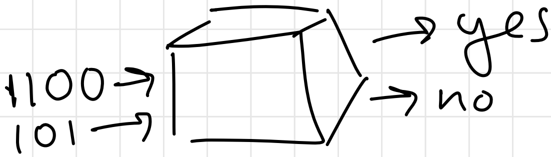
$((0+1)(0+1))^*$: all binary strings of even length



Finite-State Machines

A language we want to recognize

$\{ w \in \{0,1\}^* : |w| \text{ is a multiple of } 5 \}$

0110111100010 → Y / N

Keep track of the remainder:

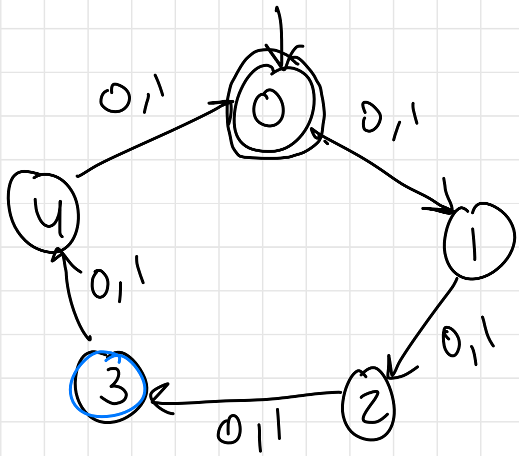
0, 1, 2, 3, 4

length mod 5

as an algorithm:

```
rem = 0
for i = 1 to |w|:
    rem = (rem + 1) mod 5
return (rem == 0)
```

as a finite-state machine:



ex

$\delta(3,0) = 4$

Deterministic Finite Automaton (DFA)

(Q, s, A, δ)

Q : set of states $\{0,1,2,3,4\}$

$s \in Q$: start state 0

$A \subseteq Q$: set of accepting states $\{0\}$

$\delta : \underbrace{Q \times \Sigma}_{\text{function takes in (state, symbol)}} \longrightarrow Q$: transition function

and returns a state

δ:

	0	1
0	1	1
1	2	2
2	3	3
3	4	4
4	0	0

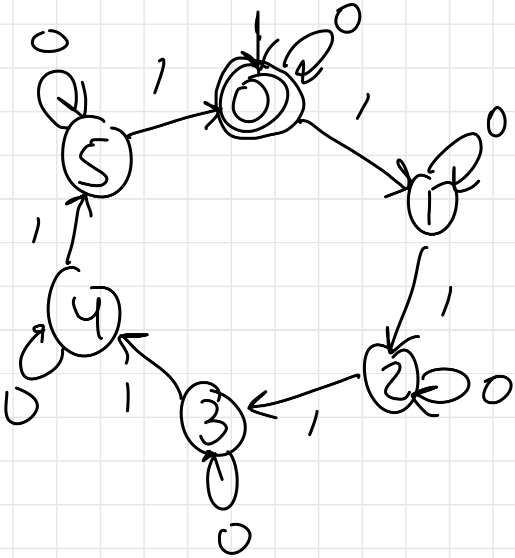
$\delta(q, a) = (q+1) \bmod 5$

binary strings where # of 1's is divisible by 6.

$Q = \{0, 1, 2, 3, 4, 5\}$

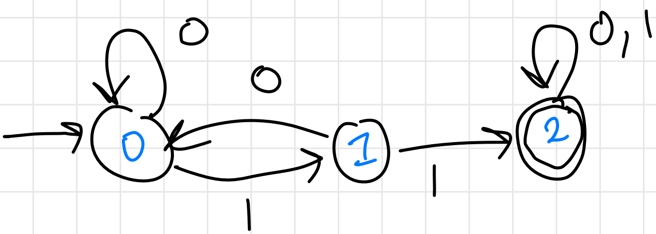
$s = 0$

$A = \{0\}$



$$\delta(q, a) = \begin{cases} q & \text{if } a = 0 \\ (q+1) \bmod 6 & \text{if } a = 1 \end{cases}$$

what strings does this DFA accept?



what do the states mean?

Y 2 : saw 11 at some point

1 : just a 1 (but haven't seen 11 yet)

0 : just saw a 0 or ε and haven't seen 11 yet

Extended Transition Function

$$\delta^* : Q \times \Sigma^* \to Q$$

string over alphabet Σ

$$\delta^*(q, w) = \begin{cases} q & \text{if } w = \varepsilon \\ \delta^*(\delta(q,a), x) & \text{if } w = ax \end{cases}$$

for above DFA:

$$\delta^*(0, 110101011000) = 2$$

For machine $M = (Q, s, A, \delta)$

$$\underbrace{L(M)}_{\text{"the language recognized by machine } M\text{"}} = \{ w : \delta^*(s, w) \in A \}$$

(somewhat) automatic way to design DFAs.

- write an algorithm
  - ✓ for loop through the symbols
  - ✓ constant # variables
  - each variable can only take constant values

```
CONTAINS11(w[1..n]):
  found ← FALSE
  for i ← 1 to n
    if i = 1
      last2 ← w[1]
    else
      last2 ← w[i−1] · w[i]
    if last2 = 11
      found ← TRUE
  return found
```

→ while

vars:
- last2
- found

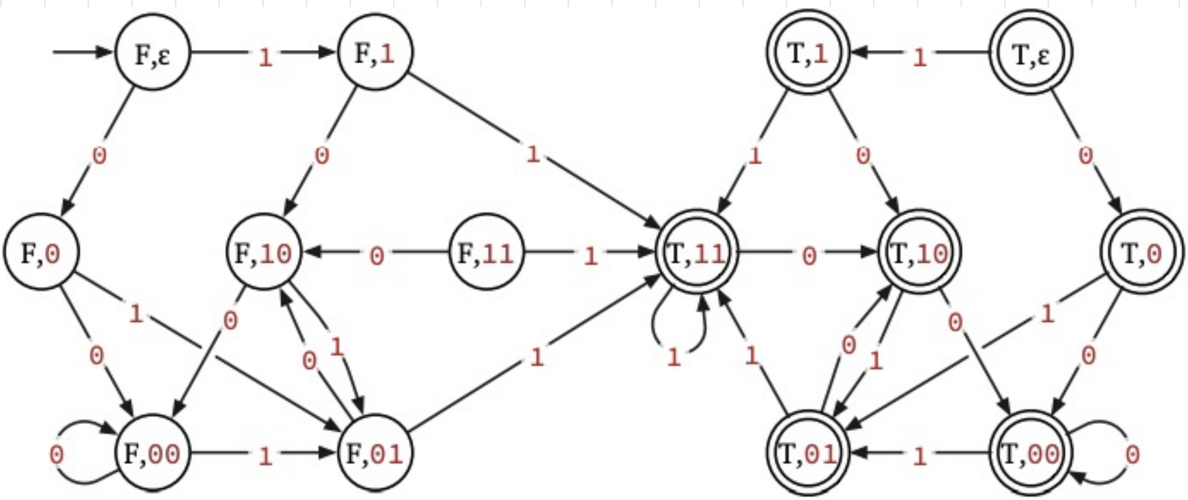
last2:
11, 10, 01, 00, 1, 0, ε

found:
T, F

DFA recognizing strings w/ substring 11

q	$\delta[q, 0]$	$\delta[q, 1]$
(False, ε)	(False, 0)	(False, 1)
(False, 0)	(False, 00)	(False, 01)
(False, 1)	(False, 10)	(True, 11)
(False, 00)	(False, 00)	(False, 01)
(False, 01)	(False, 10)	(True, 11)
(False, 10)	(False, 00)	(False, 01)
(False, 11)	(False, 10)	(True, 11)

q	$\delta[q, 0]$
(True, ε)	(True, 0
(True, 0)	(True, 00
(True, 1)	(True, 10
(True, 00)	(True, 00
(True, 01)	(True, 10
(True, 10)	(True, 00
(True, 11)	(True, 10



binary #'s divisible by 5

ex: $1010 = 0\cdot 2^0 + 1\cdot 2^1 + 0\cdot 2^2 + 1\cdot 2^3 = 2+8$
$\uparrow\uparrow\uparrow\uparrow$ $= 10$

ex: 0010111010110

i	w[1..i]	w[i]	decimal value	value mod 5
0	ε	n/a	0	0
1	0	0	0	0
2	00	0	0	0
3	001	1	1	1
4	0010	0	2	2
5	00101	1	5	0
6	001011	1	11	1
7	0010111	1	23	3
8	00101110	0	46	1
9	001011101	1	93	
10	0010111011	1	187	
11	00101110110	0	374	

$$\text{value}(w[1\cdots i]) = 2(\text{value}(w[1..i-1]) + w[i]$$

$$\left(2(\text{value}(w[1..i-1]) + w[i]\right) \bmod 5$$

```
MultipleOf5(w[1 .. n]):
    rem ← 0
    for i ← 1 to n
        rem ← (2 · rem + w[i]) mod 5
    if rem = 0
        return True
    else
        return False
```

rem:

0, 1, 2, 3, 4