with your table, discuss:

Give a DFA that recognizes binary strings divisible by 5.

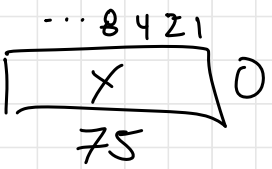
8 4 2 1
1010 is 10 in decimal accept

16 8 4 2 1
00111 is 7 in decimal reject

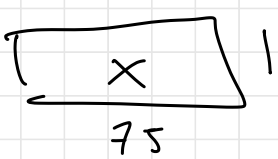
remember:
- we have to process input L to R
- we can't have unbounded variables (infinite states)

```
binaryToDivBy5(w):
  decimal = 0
  for i in len(w):
      decimal = 2decimal
                + w[i]
```

Trick:



doubled



```
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
```

draw graphical DFA

- states Q
- start ↓
- Accepting A ◎
- transitions 1 ⇉ 0 ⇊

$\delta(q, a) = (2q + a) \bmod 5$

How to go from an algorithm to a DFA

```
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
```

first character we see

last2: 0, 01, 11, 10

found: {FALSE, TRUE}

last2: {ε, 0, 1, 01, 00, 10, 11}

Q = ordered pairs of vars

(found, last2)

(TRUE, 11) $\xrightarrow{0}$ (TRUE, 10)

(TRUE, 11) $\xrightarrow{1}$ (TRUE, 11)

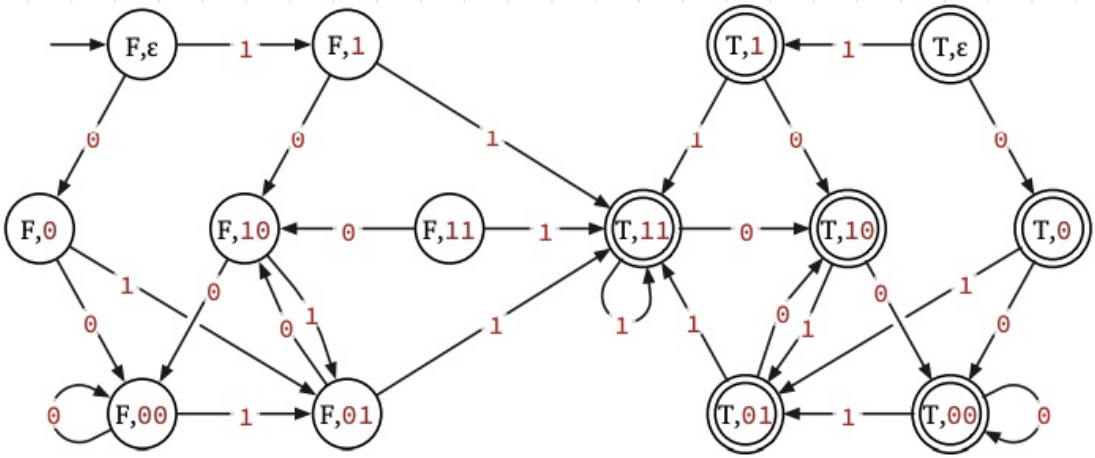
(FALSE, 1) $\in Q$

s = (FALSE, ε)

T or F?

A = any state w/ TRUE in 1st position

δ : as in algorithm

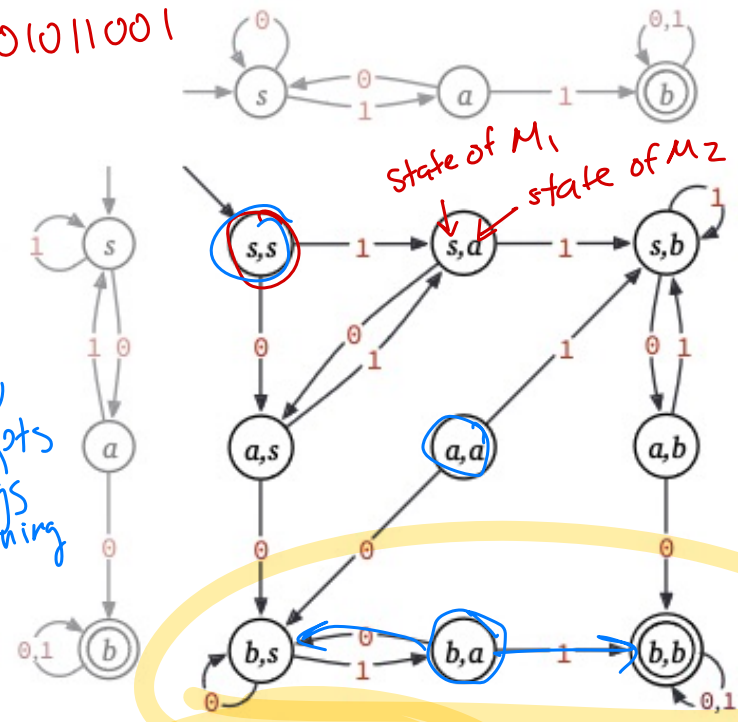

F,ε
F,1
T,1
T,ε
F,0
F,10
F,11
T,11
T,10
T,0
F,00
F,01
T,01
T,00

Product Construction

Make a DFA that accepts strings containing both substring 00 and 11.

M_2, accepts strings containing substring 11

G10101100!



M_1, accepts strings containing 00

← M

(b,a) → 1, 0 ↓

Given $M_1 = (Q_1, \Sigma_1, s_1, A_1, \delta_1)$

$M_2 = (Q_2, \Sigma_2, s_2, A_2, \delta_2)$

The product construction M:

$Q = Q_1 \times Q_2 = \{(p, q) : p \in Q_1, q, Q_2\}$

$S = (S_1, S_2)$

$A = \{(p,q) : p \in A_1 \text{ and } q \in A_2\}$

$\delta((p,q), a) = (\delta_1(p, a), \delta_2(q,a))$

$\delta_1(p,a)$: state in M_1; $\delta_2(q,a)$: state in M_2