1-1

DFA : String ⇒ Boolean

addition #1. "x + y = z" s.t. $\hat{z} = \hat{x} + \hat{y}$

"1+1=2" ∈ Add "1+2=4" ∉ Add

#2. use a transducer — transducer

:= DFA + output alphabet + output

DFA = $(Q, \Sigma, q_0, \delta, F)$

Transducer = $(Q, \Sigma, \Gamma, q_0, \rightarrow \delta', F)$

↑ output alphabet $\rightarrow \delta, \gamma, F)$

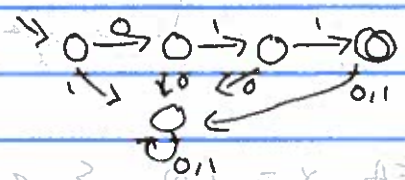
$\delta' : Q \times \Sigma \Rightarrow Q \times \Gamma$ $\quad \gamma : Q \rightarrow \Gamma$

trans : string ⇒ string

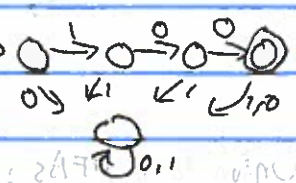
Σ Γ "1+1" → "2"

FIN ⊆ REG ⇒ Every finite has a DFA

{011}



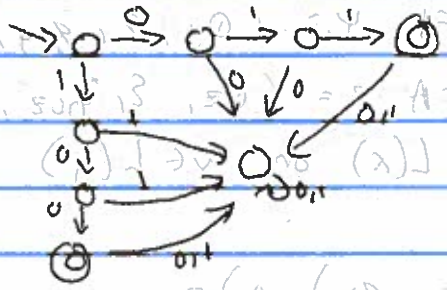
{100}



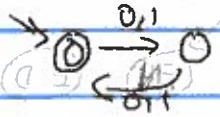
{011, 100}

= 011 or 100

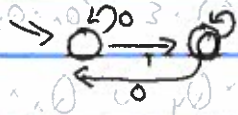
= {011} ∪ {100}



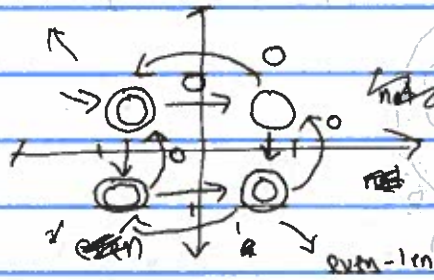
Even-length strings



odd-number strings



even-len OR odd number



3-2/ Can I take two DFAs and "or" them?

Programming: f : str → bool g : str → bool

h : str → bool

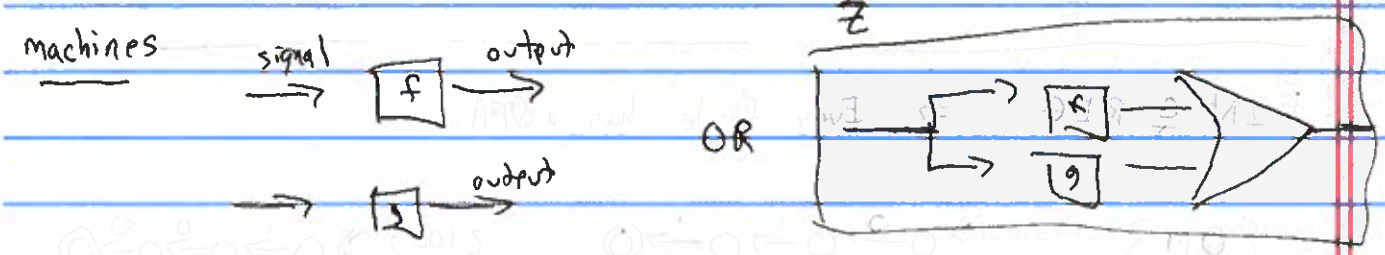
h x = true iff f x = true OR g x = true

h x = (f x) or (g x)

```
for c in x;
    update fguess
    update gguess
output (fguess) or (gguess)
```

```
f
for c in x;
    update state
output ans
```

machines



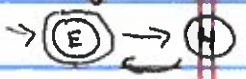
Union on DFAs: input DFA $X = (Q_X, \Sigma, q_{0X}, \delta_X, F_X)$

DFA $Y = (Q_Y, \Sigma, q_{0Y}, \delta_Y, F_Y)$

output DFA $Z = (Q_Z, \Sigma, q_{0Z}, \delta_Z, F_Z)$

$w \in L(Z)$ iff $w \in L(X)$ or $w \in L(Y)$

even-length:



= X

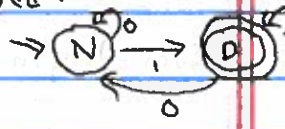
$Q_Z = Q_X \times Q_Y$

$q_{0Z} = (q_{0X}, q_{0Y})$

$\delta_Z = Q_Z \times \Sigma \to Q_Z$ $\quad = (Q_X \times Q_Y) \times \Sigma \to (Q_X \times Q_Y)$

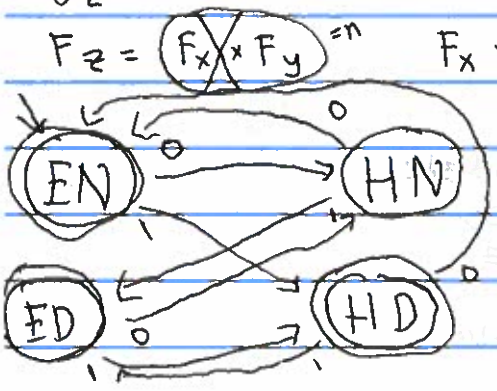
$\delta_Z((q_{iX}, q_{iY}), a) = (\delta_X(q_{iX}, a), \delta_Y(q_{iY}, a))$

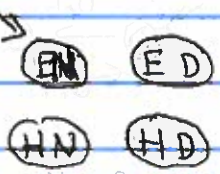
odd:



= Y

$F_Z = F_X \times F_Y$ $\quad = F_X \times Q_Y \cup Q_X \times F_Y$





= Z

3-3

Closure - property

~~Operation~~ "Set A is closed under operation F"

$\forall x \in A.\ F(x) \in A.$ (if F is unary)

$\forall x, y \in A.\ F(x, y) \in A.$ (if F is binary)

"REG is closed under union" $\quad \forall A \in REG$

$\forall X, Y \in REG.\ X \cup Y \in REG \quad = \exists D \in DFA,\ L(D) = A$

"REG is closed under intersect"

$(F_Z = F_X \times F_Y)$

"REG is closed under complement"

in $X = (Q_X, \Sigma, q_{0X}, \delta_X, F_X)$

out $Z = (Q_Z, \Sigma, q_{0Z}, \delta_Z, F_Z)$

$Q_Z = Q_X \quad q_{0Z} = q_{0X} \quad \delta_Z = \delta_X \quad F_Z = Q_X - F_X$

"The regular operations" := "the operations REG is closed under"

$\{011, 110, 0000\} = \{011\} \cup \{110\} \cup \{0000\}$

$\left(\left(\boxed{\text{even 0s}} \cup \boxed{\text{ends in 01}} \right) \cap \boxed{\text{has less than 4 1s}} \right)^C$

"even-len" "ends in 01" "has less than 4 1s"

"Strings that are even until there's a 01 and then end in a 1"

even-len | 01 | odd-num

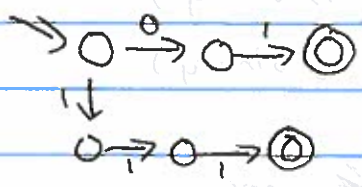
concat $(x \circ y) \quad (A \circ B)$ $\quad \wedge \leftarrow$ ~~AA~~ $\in$ AND

$w \in A \circ B$ iff $w = w_A w_B \wedge w_A \in A \wedge w_B \in B$

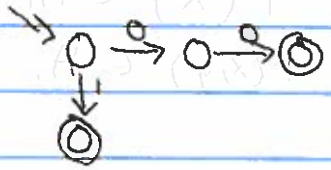
$A = \{011, 110\} \quad B = \{000, 1\} \quad A \circ B = \{011000, 0111, 110000, 1101\}$

3-4

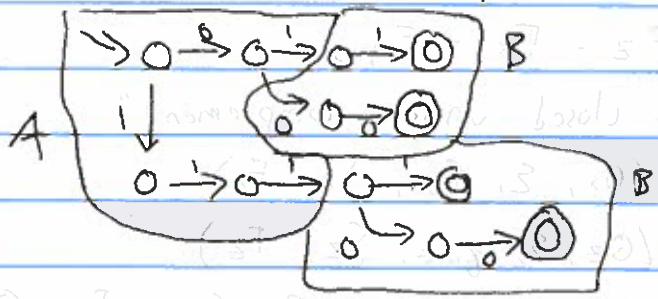
$A = \{01, 111\}$



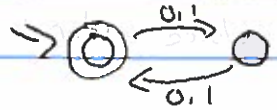
$B = \{1, 00\}$



$A \circ B = \{0111, 0100, 1111, 11100\}$



C = even



$C \circ B$ = even after that 1 or 00

= ???

$\varepsilon \in C$	$1 \in C \circ B$	$00 \in C \circ B$
$01 \in C$	$011 \in$	$0100 \in$
$11 \in C$	$111 \in$	$1100 \in$

"All binary strings where the third character from the end is 1"