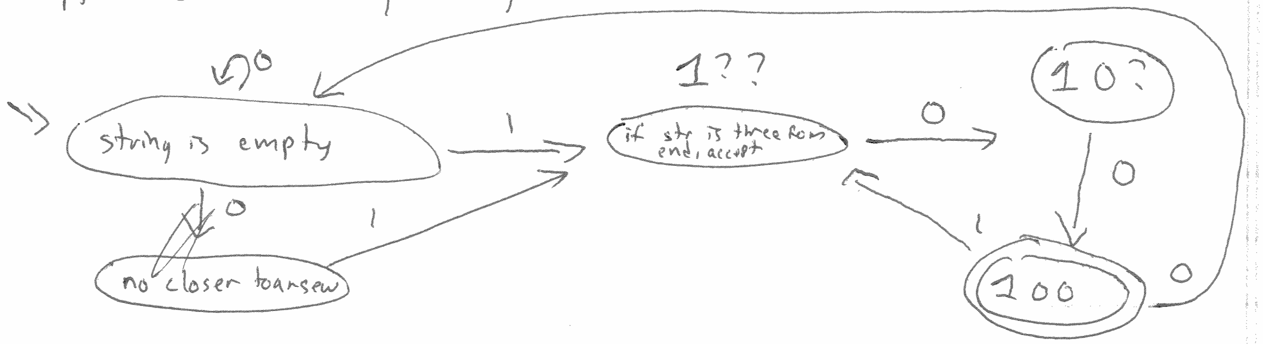
3-4/

```
typedef enum { XL, YM, YN, ZN} state_t;

int machine ( )   {

    state_t  st  = XL;
    while ( char c = getc() ) {
      switch (st) {
      case XL:  switch (c)  {
                  case '0' :  st= YM;   break;
                  case '1' : st = ZN;   }
      case YM :  st = YN;  b;
     case YN :  st = YN; b;
    case ZN :  st = ZN; b; } }
  return  st == YN;
```

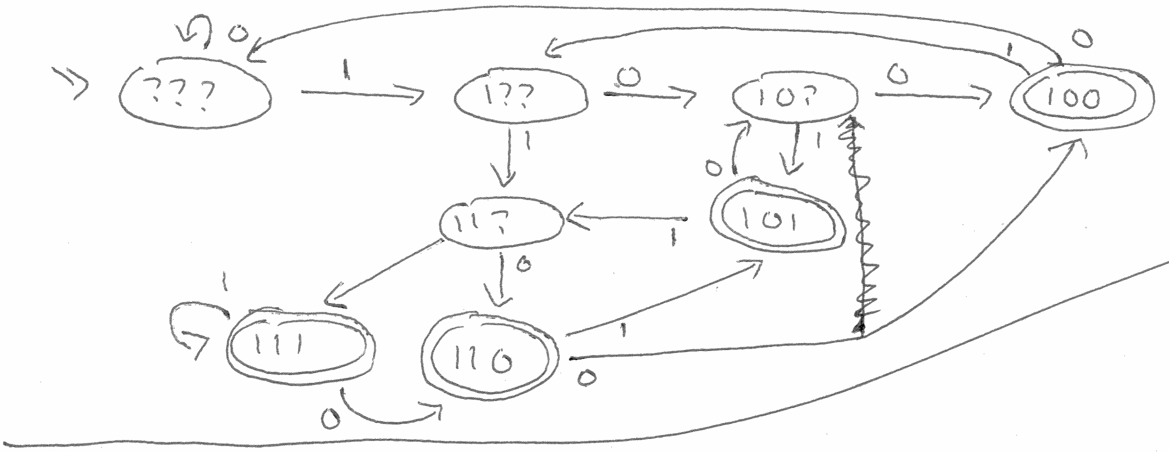
Regexp : $\underbrace{\text{courses /}}_{x}\underbrace{*}_{any^*}\underbrace{\text{/ grades /}}_{y}\underbrace{*}_{any^*}\underbrace{\text{, fail}}_{z}$

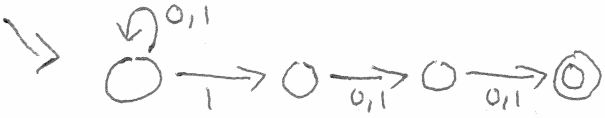
$re = x \circ \Sigma^* \circ y \circ \Sigma^* \circ z$

1-1/ $A = \{$ all binary strings where third-to-last character is $1 \}$



0100 ✓
1000 X
11100 ✓





NFA -

Non-deterministic Finite Automata

diff 1 : states don't have all transitions

diff 2 : states may have >1

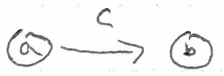
states can any number of trans. per character

Future telling

backtracking

fork()-ing

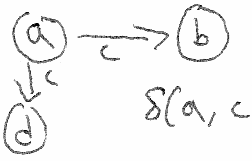
mystery good one



$\delta(a, b) = c$

$$\text{DFA } d = \langle \Sigma, Q, q_0 \in Q, \delta : Q \times \Sigma \to Q, F \subset Q \rangle$$

$$\text{NFA } n = \langle \Sigma, Q, q_0 \in Q, \delta : Q \times \Sigma_\varepsilon \to P(Q), F \subset Q \rangle$$

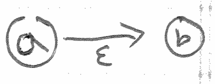


$\delta(a, c) = \{b, d\}$

$\delta(a, c) = \{\}$

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

epsilon



$\delta(a, \varepsilon) = \{b\}$

4-2

$L(\text{NFA } n) = \{ x \mid x \text{ is accepted by NFA } n \}$

A string x is accepted by NFA n iff

$q_0 \overset{x}{\Rightarrow}{}^* q_i$ s.t. $q_i \in F$ [exactly the same as DFA

An NFA n runs from q_i to q_j on x $(q_i \overset{x}{\Rightarrow}{}^* q_j)$

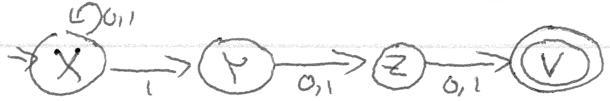
$q_i \overset{\varepsilon}{\Rightarrow}{}^* q_i$ $q_i \overset{ax}{\Rightarrow}{}^* q_k$ iff $q_i \overset{a}{\Rightarrow} q_j$ where $q_j \in Q$

$q_j \overset{x}{\Rightarrow}{}^* q_k$ $a \in \boxed{\Sigma_\varepsilon}$

only difference from DFA

An NFA n steps from q_i to q_j on $\underbrace{~~x~~ a}_{\Sigma_\varepsilon}$ $(q_i \overset{a}{\Rightarrow} q_j)$ (for DFA $= \Sigma$)

$q_j \underbrace{\in}_{= \text{ for DFA}} \delta(q_i, a)$



$\xrightarrow{1}$ [Y] 100 $\overset{1}{\Rightarrow}$ [Z] 00 $\overset{0}{\Rightarrow}$ [V] 0

[X] 11100 $\overset{1}{\Rightarrow}$ [X] 1100 $\Rightarrow$ [X] 100 $\Rightarrow$ [X] 00 $\Rightarrow$ [X] 0 $\Rightarrow$ [X]

$\Rightarrow$ [Y] 00 $\Rightarrow$ [Z] 0 $\Rightarrow$ [

$\Rightarrow$ [Y] 1100 $\Rightarrow$ [Z] 100 $\Rightarrow$ [V] 00

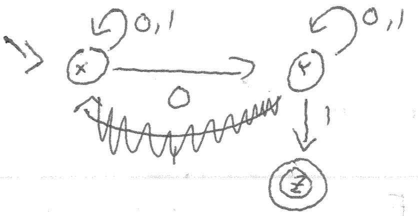
$Y \in \delta(X,1) = \{X, Y\}$

$X \overset{11100}{\Rightarrow} V$? ✓ $11100 \in L(n)$

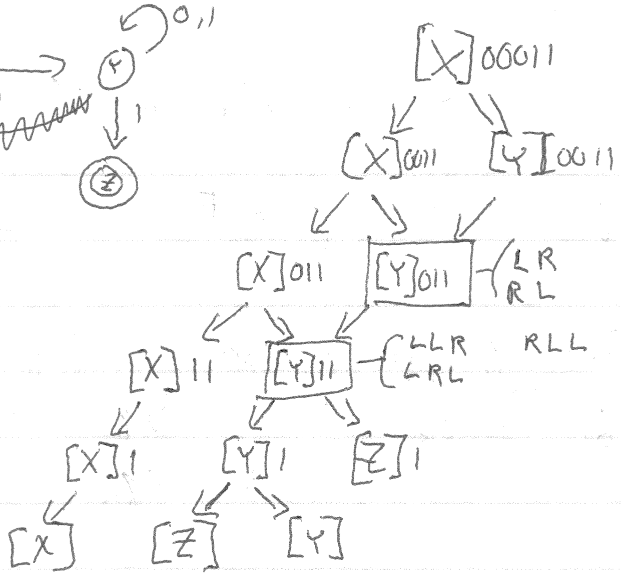
back-tracking = DFS oracle = cheating

fork()-ing = BFS

1-3/



00011



[{X}] 00011

↓

[{X,Y}] 0011

↓

[{X,Y}] 011

↓

[{X,Y}] 11

↓

[{X,Y,Z}] 1

↓

[{X,Y,Z}] ✓