CS 373

Lecture #2

Strings

Σ - alphabet (a finite set)

$$\Sigma = \{a, b, c, d, \ldots z, A, B, C, D, \ldots Z\}$$

A string/word over the alphabet Σ is a finite sequence of elements of Σ

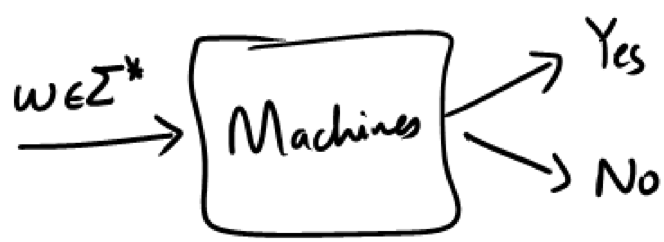
Eg. $\Sigma = \{a, b, c\}$

example string over Σ $\doteq$ a, b, b, a, c

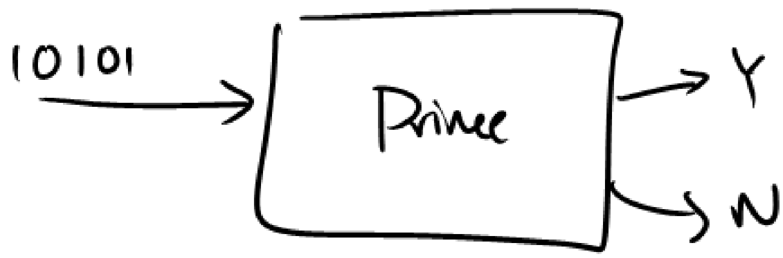
$\doteq$ abbac

String : $[1, n] \rightarrow \Sigma$.

Σ



Our programs work on strings:



ASCII — Alphabet

ASCII file - word over the ASCII alphabet

Obama ␢ is ␢ the ␢ \n president

Notations on strings.

Length of a string: # of characters in it.

Empty word: the word with no characters in it

ϵ

$|w|$ = length of string w

$|\epsilon| = 0$

Concatenation

$\Sigma = \{a, b, c\}$

$aaba \circ bacc = aababacc$

$bacc \circ aaba = baccaaba$

$$w \circ \epsilon = \epsilon \circ w = w$$

Prefix w is a prefix of x if $\exists y \in \Sigma^* .\ w \circ y = x$

Eg. abb is a prefix of $abbab$

Is abb a prefix of abb?

Yes (you can choose $y = \epsilon$).

<u>Note</u> ϵ is a prefix of x, for any x.

w is a prefix of w, for any w.

__Suffix__ w is a suffix of x

if $\exists y \in \Sigma^* .\ y \circ w = x$

- w is a suffix of w, $\forall w$
- ϵ is a suffix of x, $\forall x$.

__Subsequence__/substring

w is a subsequence of x if

$\exists y, z \in \Sigma^* .\ y \circ w \circ z = x$.

Often (almost always), we'll drop "$\circ$".

$$y \circ z = yz$$

$5x$

$5 * x$

$$x \circ (y \circ z)$$
$$= (x \circ y) \circ z$$
$$= xyz$$

$\Sigma = \{a, b, c\}$ $\quad x = ab \quad y = bc$

$xy = abbc$; ab ; ax

Language

Σ

$$\mathcal{L}(M) = \{ x \in \Sigma^* \mid M \text{ accepts } x \}$$

By def., for any word $y \notin \mathcal{L}(M)$, M does not accept y.

A language L over Σ is
a subset of Σ^*.
($L \subseteq \Sigma^*$; i.e. L is
a set of
words
over Σ).

$\Sigma = \{0,1\}$

$L_{primes} = \{10, 11, 101, \ldots\}$

Implicitly describing languages

$$X = \{ 2y \mid y \in \mathbb{N} \}$$

$$L = \{ x \in \{a,b\}^* \mid |x| \text{ is even} \}$$

$$= \{ \epsilon, \; aa, ab, ba, bb, \; aaaa, \ldots \}$$

Finite automata

"Machines with constant amount of memory"

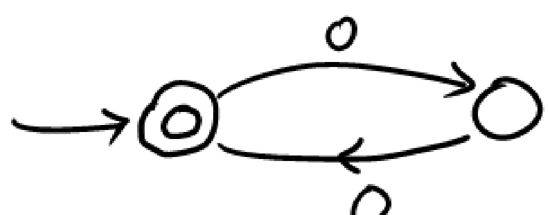
$$\Sigma = \{0\}$$

$$L = \{ w \in \{0\}^* \mid |w| \text{ is even} \}$$

```
main() {
  n: count of length of |w|
  if (n is even) say "yes"
           else say "no"
}
```

```
main ( ) { n:=0
  while (there is input to read){
    read a;
    n := ((n+1) mod 2)
  }
  if n=0 say "yes" else say "no".
```

M

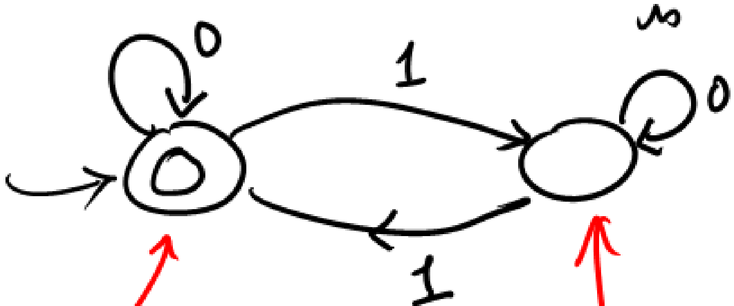


$$\mathcal{L}(M) = \{ w \in 0^* \mid |w| \text{ is even} \}$$

Example

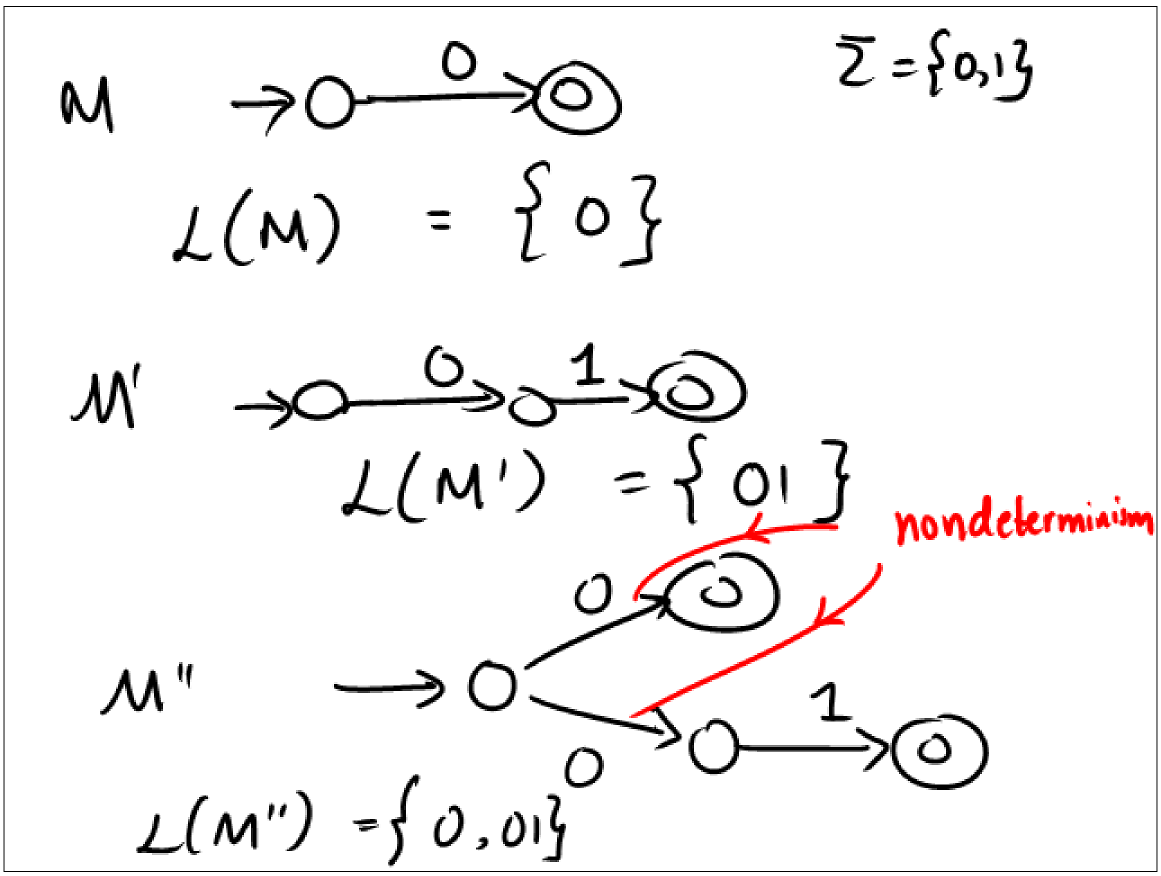
$\Sigma = \{0, 1\}$

$L = \{ w \in \Sigma^* \mid$ the number of 1's in w is even $\}$



no. of 1's seen is even

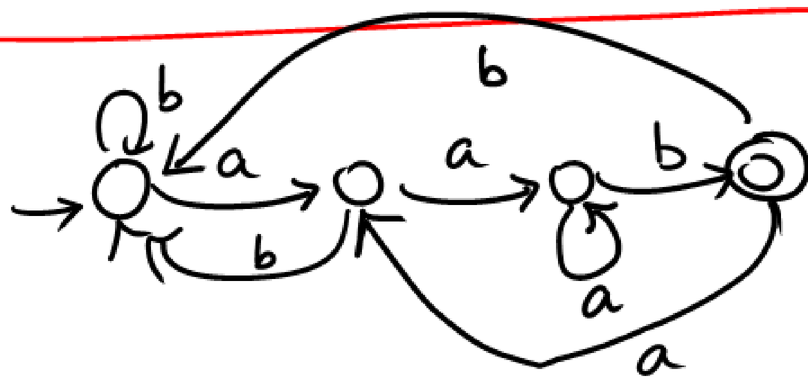
no. of 1's seen is odd


M
0
Σ = {0,1}
L(M) = {0}
M'
0
1
L(M') = {01}
nondeterminism
M''
0
1
0
L(M'') = {0, 01}

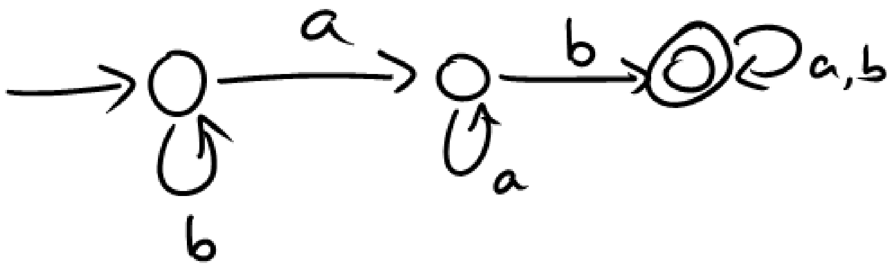
$L = \{ w \mid w \text{ ends in } aab \}$

$\Sigma = \{a, b\}$

Nondet



M



$L(M) = \{ w \mid w \text{ has a subsequence } ab \}$

Does M accept

$L = \{ ab, abab, ababab, \ldots \}$?

$aab \notin L$ No!

M 

$\Sigma = \{0,1\}$

$L(M) = \{0,1\}^*$

It is <u>absurd</u> to say M accepts L_{prime}!