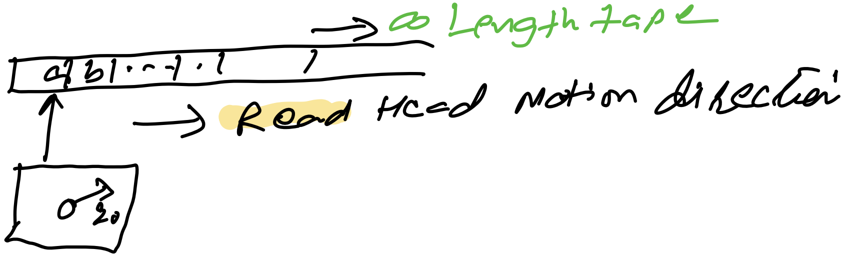
07-09-2021 Finite automata & Morphological Parsing → To segment a word into its consitnents

FA, $M = (Q, \Sigma, \delta, s, F)$, $\delta: Q \times \Sigma \to Q$

$F \subseteq Q$, s - start state



we simulated the FA through an algorithm:

data structures used: transition table[x,y]

Tape, $\underbrace{\text{Tape}[i]\ \text{Tape}[i] \ldots}_{= \text{i/p } w}$

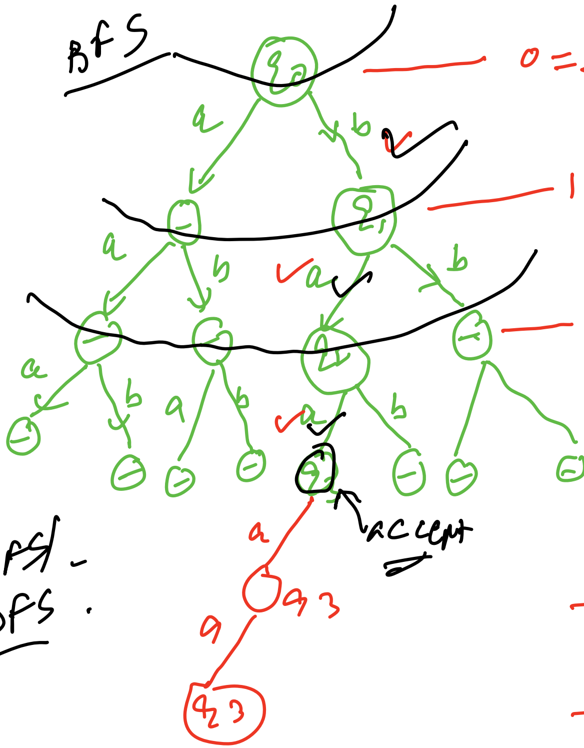
index = head position on tape.

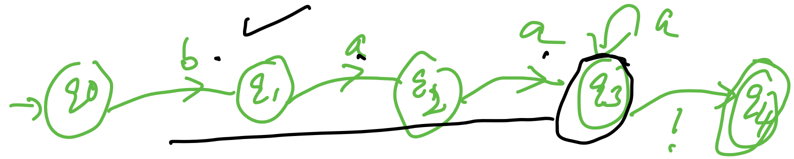
advancing head: index++.

or index = index+1

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

BFS





baa ?

$0 = 2^0 = 1$

$1 = 2^1 = 2$

$2 = 2^2 = 4$ if i/p $= |baaaa| = 5$

$3 = 2^3 = 8$ = breadth of tree at level 3

No of paths = 8,

i/p size = 3, paths $= 2^3 = 8$

$4 = 2^4 = 16$

$5 = 2^5 = 32$

if i/p size/length = 5,

no. of paths = 32
$= 2^5$

{ BFS
DFS.

If Input is of length n, $|\Sigma| = 2$,
then possible paths you need to search $= 2^n$ paths

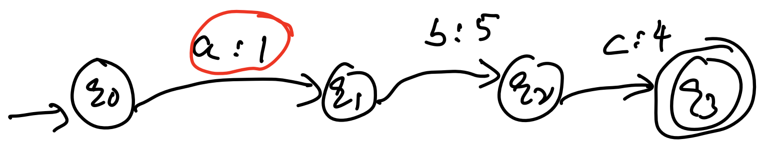
If $|\Sigma| = m$, then possible
worst case searches $= m^n$
— Exponential !!

Since BFS & DFS are complex processes:

BFS	DFS	
m^n	m^n	Exponential (Time)
m^n	$m \times n$	Space

∴ better algorithms are required: these are A^*, best first search, simulated annealing, ...

FST : Finite State Transducer



i/p = abc
o/p = 154

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

$Q = \{q_0, q_1, q_2, q_3\}$

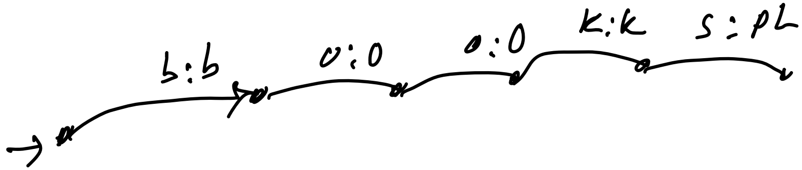
$\delta : Q \times \Sigma \to Q$

$\delta(q', \underbrace{i:o}) = q_j$

$\Sigma = ?$ $\quad I = \{a, b, c, d, e\}$
$O = \{1, 5, 4, 7\}$

$\Sigma \subseteq I \times O$, $\subseteq$ is because all the combinations of I & O are not allowed

Where to use FST? Applications:

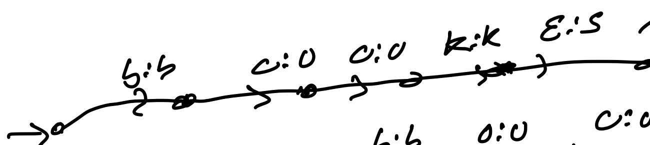


i/p = books

o/p = book+PL, the "books" has been parsed into "Book+PL"

① ↑ This is called Morphological parsing ✓

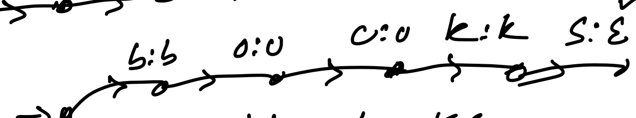
② Converting words from singular to plural



i/p = book
o/p = books

ε = null

Plural to singular.



i/p = books
o/p = book ✓

= Word inflection

③ Simple word translations: colour → color

behaviour → behavior

British English   US English

I/P c o l o u r

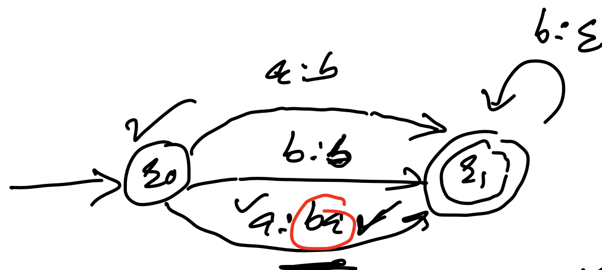
↓

O/P c o l o ε r

(null)

↑ All above are called sequential Transducers

Example



Σ^*

↑ string-to-string Transducer

No. of strings = 2

i/p = 1 string

o/p = 1 string

2- subsequential

$m = (Q, \Sigma, q_0, \delta, F, \Delta, \sigma)$

Δ ← o/p alphabet

δ ← i/p functn

σ ← o/p function

$\delta : Q \times \Sigma \rightarrow Q$

$\sigma : Q \times \Sigma \rightarrow \Delta^*$

δ, σ : partial functions

In general: p-sub-sequential Transducers