

Algorithms

---

for automata

# Applications

• grep

• speech, NLP, IR

• compilers

• Find a single pattern

• Text stream  $\rightarrow$  token stream

This book was written by  
Marc Snur . . . . .

→ [This] [book] [was] . . . . .

[c] [main] [;]

Normalize tokens

Rewrite token stream

as new token stream

- case  $\rightarrow$  lowercase [Book]  $\rightarrow$  [book]
- remove hyphens [foo-bar]  $\rightarrow$  (foobar)
- break up long tokens
  - k-digit #  $\rightarrow$  4 digit #'s



[writing]

→ [write][ing]

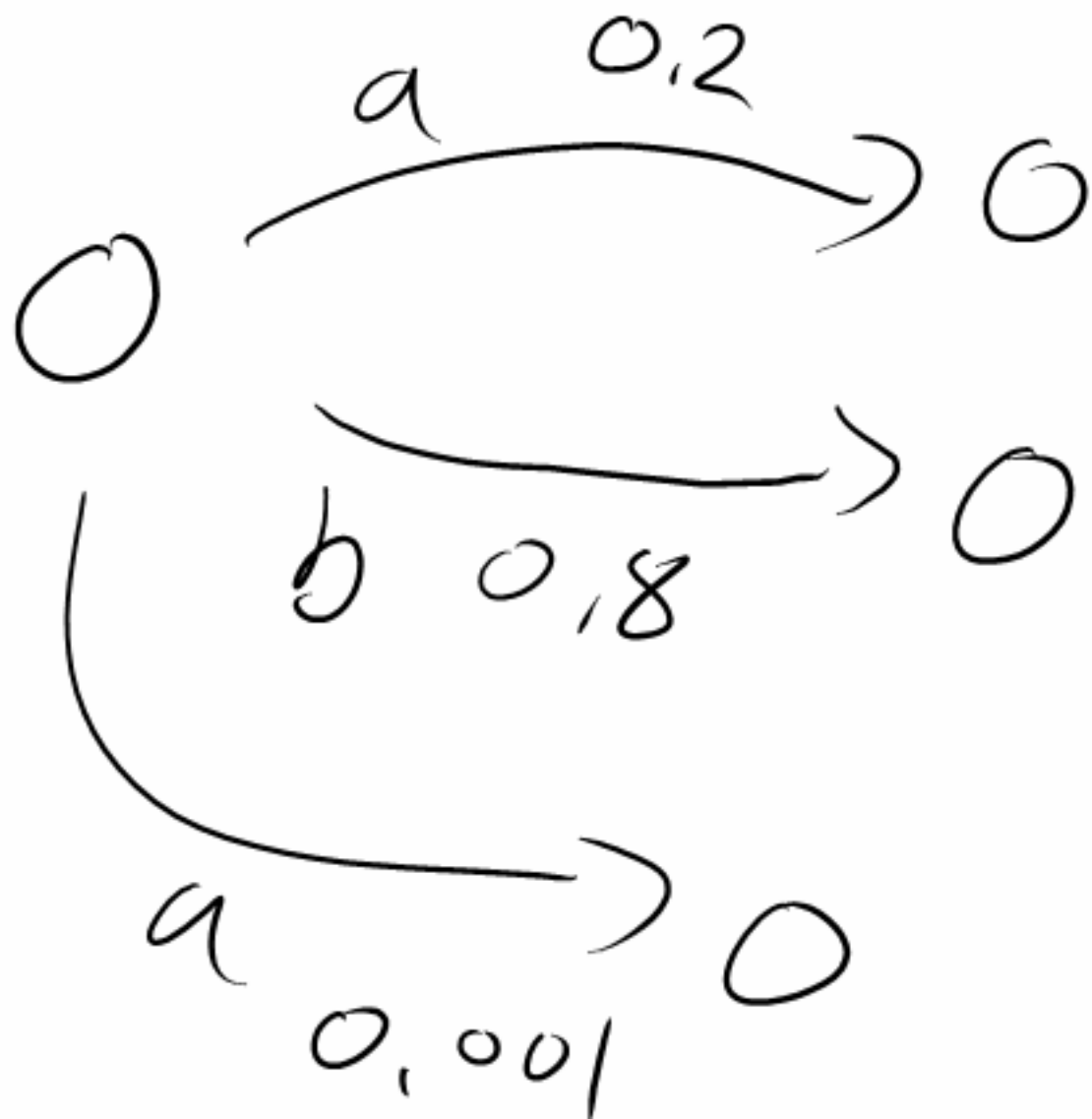
[write+ing]

$e \rightarrow \varepsilon / \_ + \overset{\wedge}{i}ng$



token rewriting rules

→ FST



• chars  $\rightarrow$  tokens

\* token rewriting

• parsing CFG

(The snow) (is falling) (in Iowa)

① regex, CS rewrite rules  
statistic transition

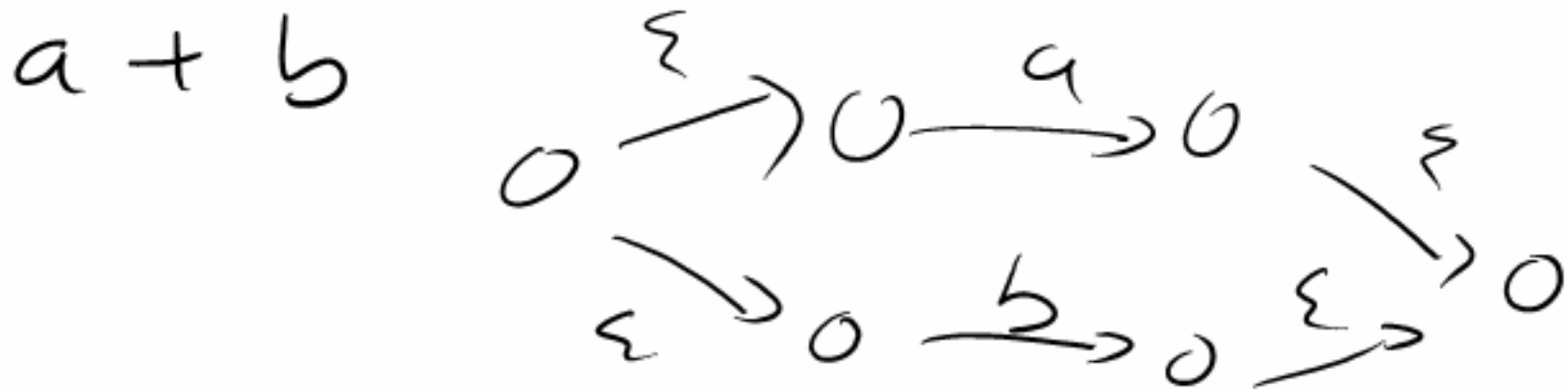
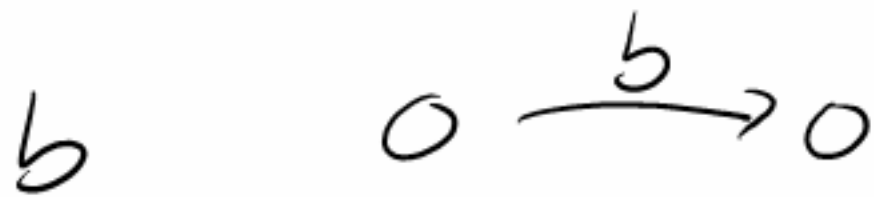
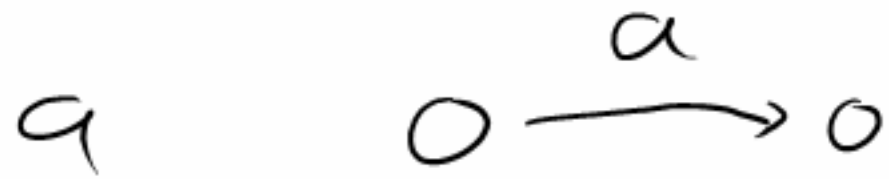
→ NFA / DFA / FST

→ run on input

② CFG  
→ optimize a bit  
→ use directly

# Formal conversion

regex  $\rightarrow$  NFA



n char in regex  $\rightarrow$  2n states in NFA

remove  $\epsilon$ -transitions

NFA  $\rightarrow$  NFA w/out  $\epsilon$

$\forall$  state, check out each transition,  
follow chain of transitions

$S$  states in NFA

$O(S^3)$  time to do it

[ for  $S$  states  
follow  $S$  transitions  
upto path length  $S$

NFA  $\rightarrow$  DFA

$s$  states in NFA

$2^s$  states in DFA

Time:

For  $2^s$  sets of states in NFA

$O(s^3)$  [ for each state in set  
find all transitions  
to other states ]  $O(s^2)$

total work  $O(2^s s^3)$

3 options

- Convert to DFA  
recognize w/ DFA

- run NFA directly on input

- build DFA as needed

run DFA ( $s$  states)

on input of length  $w$

time  $O(w)$  or  $O(w \log s)$

run NFA on input if in theory

$O(ws^2)$

a) CS rules  $\rightarrow$  FST

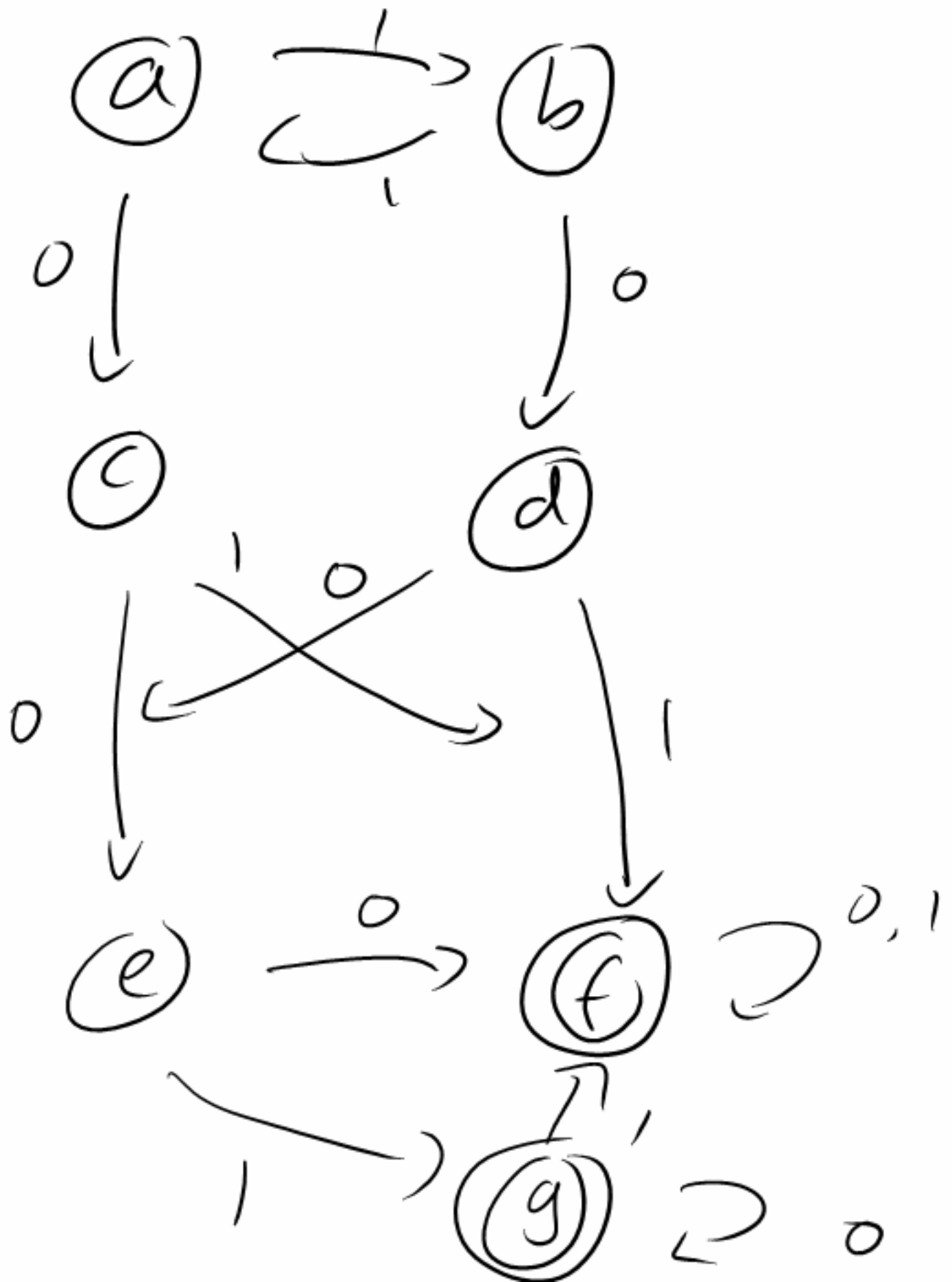
b) DFA (NFA)  $\rightarrow$  regex

O (s<sup>3</sup> 4<sup>s</sup>)

ICK

↗ Cat  
↘ call  
↙ cost

→ c → 0 → a →  $\frac{+}{ll}$   
↘ 0 →  $\frac{}{ST}$



$p$  &  $q$  are DISTINCT

a) if  $p \in F$ ,  $q \notin F$

b) if  $\delta(p, a)$  and  
 $\delta(q, a)$

are DISTINCT



$S$  &  $E$   
are already  
known  
DISTINCT

b							
c	1	1					
d	1	1					
e	0	0	0	0			
f	ε	ε	ε	ε	ε		
g	ε	ε	ε	ε	ε		
	a	b	c	d	e	f	

A handwritten diagram showing a grid with rows labeled b, c, d, e, f, g and columns labeled a, b, c, d, e, f. The grid contains numerical values (1, 0) and the Greek letter epsilon (ε). A large arrow points from the top right towards the grid, and a smaller arrow points from the top left towards the grid.

$n^2$  pairs of states

$\leq n^2$  iterations of loop

$\leq n^2$  work in each

---

$O(n^4)$

really can be  $O(n^2)$

or even  $O(\log n)$

Algorithm to fill in table  
of  $(p, q)$  pairs  
as DISTINCT

- if  $p \in F, q \notin F$   
mark  $(p, q), (q, p)$  DISTINCT

- loop until no change  
for each pair of states  $p, q$   
for each character  $a$

  - if  $\delta(p, a)$  and

  - $\delta(q, a)$  are

  - listed as DISTINCT

  - mark  $(p, q)$  as DISTINCT