What languages can be decided by programs that use only finite prespecified amount of memory?

→ any finite language can be → but not just finite languages

↳ finitely many input that are prescribed "yes"

equiv. question

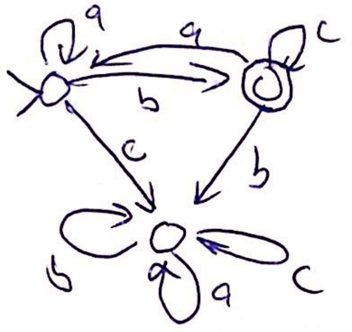
→ what languages can be decided by finite state machines? FSM

Kleene's theorem: A language is decidable by some FSM iff it can be described by a regular expression

↓

it is regular

Finite State Automata (FSA)



◎	accept (yes) state
○	state
>○	start state
$\circ \xrightarrow{a} \circ$	transition under reading a

FSA

What language does this machine accept? $\Sigma = \{a, b, c\}$ alphabet

~~bc*~~ $b\, c^*$? No

$b\,(a \vee \lambda)$ ---- ?

$(a \vee bc^*a)^*\, b\, c^*$ is proposed live by Prof.

a|a|ba|bcca|bcccc

An FSA is a five tuple (S, s_0, Σ, f, F)

where S is a finite set (of states), non empty

$s_0 \in S$ the start state

Σ is a finite alphabet

$f : S \times \Sigma \longrightarrow S$ transition function

current state — character read next — next state

$f(p, a)$ is the state reached from state p when reading a.

$F \subseteq S$ accepting states

An FSA configuration is a pair in $S \times \Sigma^*$, i.e.

a pair of a state p and a remaining input string w

(p,w) is a configuration when $p \in S$ and $w \in \Sigma^*$

when computing on input string w, the starting configuration is (s_0, w)

For any $a \in \Sigma$, string $w \in \Sigma^*$, and state $p \in S$,

the configuration (p, aw) transitions to $(f(p,a), w)$

(q,w) when $q = f(p,a)$ written $(p, aw) \vdash_M (f(p,a), w)$

one step of FSA computation

for FSA M.