Lecture 2 DFA

Tuesday, January 12, 2021   9:46 AM
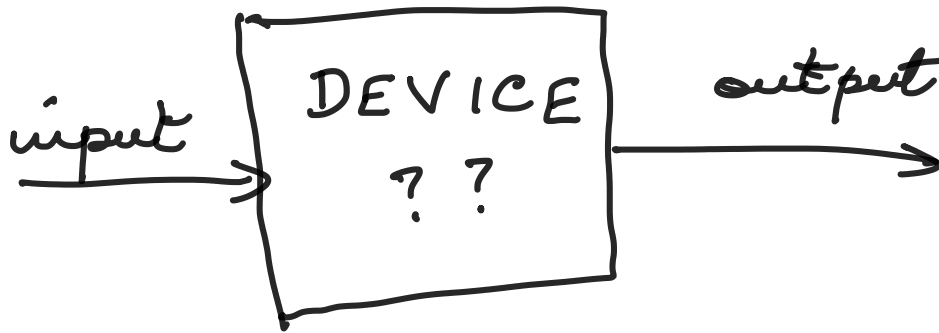
FIRST MODEL:

Finite memory machines

Finite state machines

input → sequence of symbols

collection of symbols : *alphabet*

each symbol : *letter*

Σ : alphabet

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

$\Sigma = \{0, 1\}$ or $\Sigma = \{0, 1, 2, \cdots 9\}$

words or *strings* are sequences

of letters: $\Sigma = \{a, b\}$

$a,\ abba,\ baba,\ aaaaab,\ \varepsilon,$

Σ^* : set of <u>all</u> possible $\underbrace{\text{empty word}}$ words

If $\Sigma \neq \emptyset$ then Σ^* is infinite as a set. We do <u>not</u> have infinitely long words.

$L \subseteq \Sigma^*$

$\hookrightarrow$ a <u>language</u>.

L can be finite or infinite.

Our DFA are going to read words one symbol at a time from left to right with <u>no</u> <u>backtracking</u>. At the end the DFA is going to output "accept" or "reject".
The machine is telling you a set of words that it accepts i.e.

is defining a language.
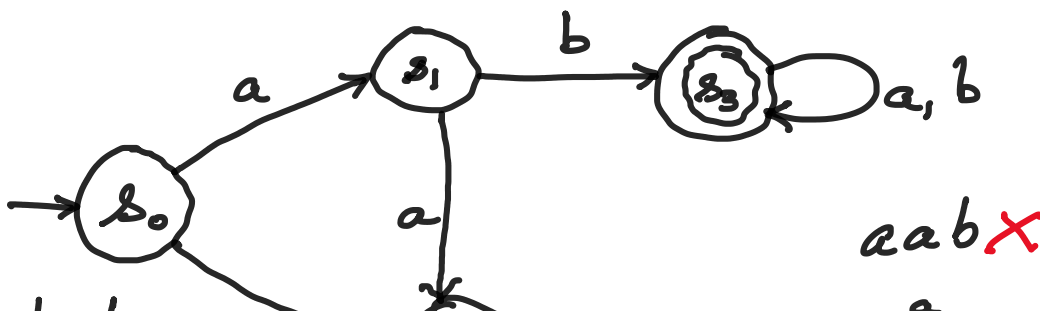
Q: Is it possible to design
a DFA for every language?

A: NO!

MACHINES :

States (finitely many)

transitions : jump between
states

We will give a formal def.
soon but let's draw pictures
first :

aab ✗

start
state b $\longrightarrow$ (S_2) $\circlearrowleft$ a, b

$S_0 \xrightarrow{} S_1 \xrightarrow{} S_2 \xrightarrow{0} S_2$

$a \quad ba \quad \checkmark$

$S_0 \xrightarrow{a} S_1 \xrightarrow{b} S_3 \xrightarrow{a} S_3$

set of states: S_0, S_1, S_2, S_3

start state: S_0

transitions are shown as
labelled arrows:

for every letter from every
state there is 1 arrow.

$S_2 \xrightarrow{a} S_3 \qquad S_2 \xrightarrow{b} S_1$

Only one arrow for each letter

the machine is deterministic.

The state with the double
circle is an accept state.

If the m/c ends up in an accept
state at the end of the string, the
string is accepted.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

OneNote   2021-01-12, 1:03 PM

https://cac-onenote.officeapps.live.com/o/onenoteframe.aspx?ui=en%…directionreason=Force_SingleStepBoot&rct=Medium&ctp=LeastProtected   Page 4 of 11

This machine recognizes all words
that start with ab

FORMAL DEF:

S: set of states (finite)

$$ \overset{a}{s \longrightarrow t} $$

$\delta(s, a) = t$

$\delta : S \times \Sigma \longrightarrow S$

$\hookrightarrow \delta$ is a (total) function

Def A Deterministic Finite Automaton

(DFA) is a 4-tuple

(S, s_0, δ, F) where

S is a finite set of states

$s_0 \in S$ is the <u>start</u> state

$\delta : S \times \Sigma \longrightarrow S$ is the

transition function

$F \subseteq S$ accept states

(final states)

Def A language that can be

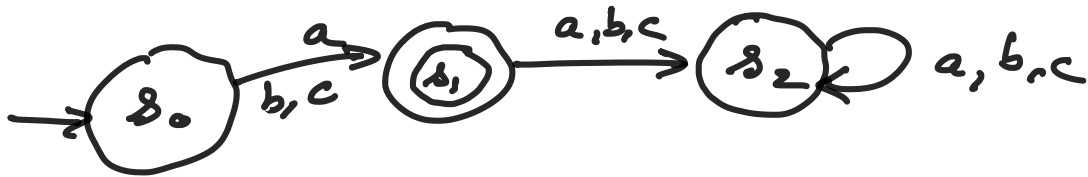recognized by a DFA is called

a <u>regular</u> language.

Assignment Project Exam Help

REMARK https://powcoder.com When I ask you to design

a DFA for L it must accept

every Add WeChat powcoder word in L <u>and</u> reject

every word that is <u>not</u> in L.

a   b

s_0

It accepts all of Σ^*.

If the start state is an accept

state then _and_ _only_ then
ε is in the language.
$\Sigma = \{a, b, c\}$

COMBINATION LOCK

$$\Sigma = \{1, 2, 3, 4, 5\}$$

45213 → only valid comb.
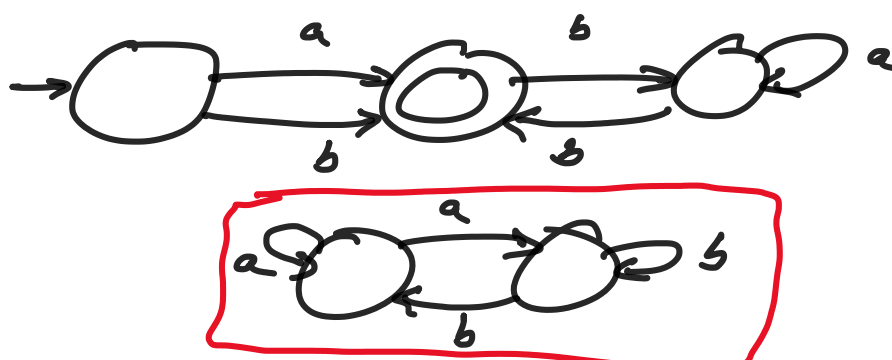
DEAD
STATE

1, 2, 3, 4, 5
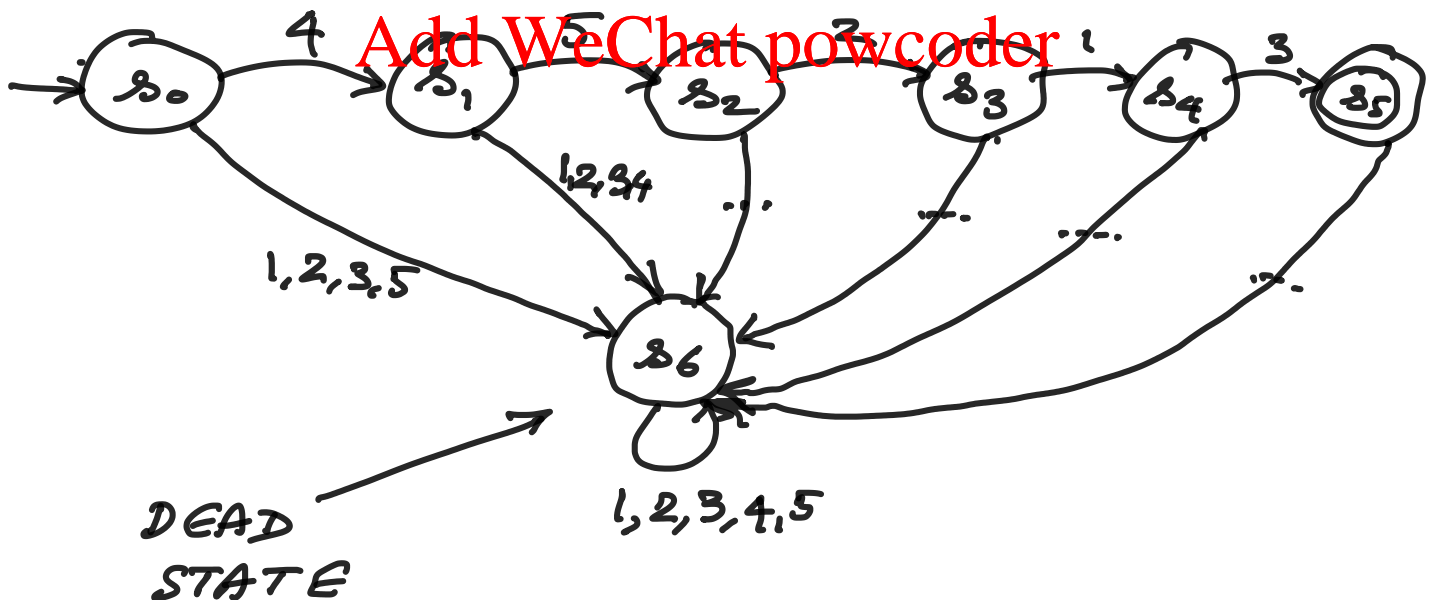
In order to get a less cluttered

picture we may leave out
the dead state and the arrows
to it. If we see a "missing"
arrow we understand that
there is actually a dead state
with the transitions hidden.

REMARK ... unreachable ... computes
They are ... but ... useless so
we usually remove them.
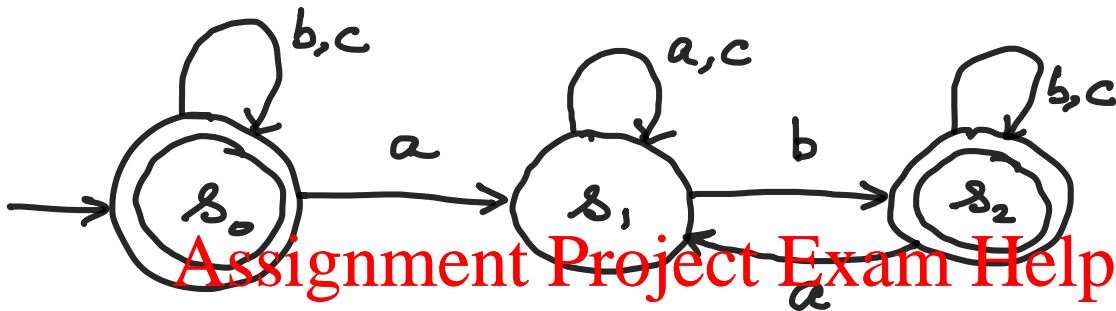

UN REACHABLE

WE WILL REMOVE THEM.

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

EXAMPLE   Design a m/c s.t it

accept only those strings s.t every
'a' is eventually followed by 'b'.

aaaccaabccb is OK

ababa ✗ not OK

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

we will interpret the strings
as binary numbers

$\varepsilon \longrightarrow 0$

$000 \longrightarrow 0$

$101 \longrightarrow 5$

Please design a DFA s.t it

accepts a sequence corresponding

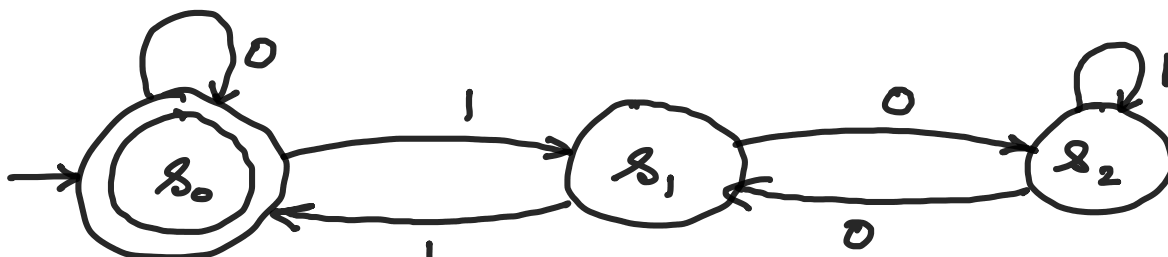... ... ... to a binary number that is <u>divisible</u> by 3.

We are reading the string from left to right.

Suppose the number you have seen so far is n

— then you read 0, what is the new number? $2n$

or then you read 1, what is the new number? $2n+1$

The states of our DFA will encode the mod 3 value of the number seen so far

EXAMPLE of a non-regular language

$$\Sigma_1 = \{a, b\}$$

$$L = \{a^n b^n \mid n \geq 0\}$$

a^n means 'a' repeated 'n' times

$\varepsilon, ab, aabb, aaabbb, \ldots$

FINITE AUTOMATA

CANNOT COUNT!

They can only count mod some finite number.

If L is finite is it necessarily regular? YES!!

PROVE IT!