SHAKTI

Dar*ts 2*6/8/1344

\* Finite Automata with O*utput \**

10

*Que. Explain Einite Automata with output*

S A *finite automata is used as accept*ar frecer *of the language. There are two different model for f*inite *automata with output capabilities. The models for finit*e *automata has output capabilities but the limitations of that model is that the output is restaicted to binary signal. Th*e models w*hich we discuss have output which i*s se*l*ected from

some o*ther alphabet this model*s are -

D Moore Machine. © Med*ly Machi*ne.

Moore Machine :

A moore *machine is defined b*y 6 tuple.

M = (Q, 2, 4, 6, 2, 20) *where,*

G = f*inite set of internal state.* z = *finite set of input alphabet.* 4 = It n*epr*e*sent theoutput alphab*et. d = *Mapping Function* 2 = *It is a mapping from a to A giving the outpu*t 90 = *Initial stat*e,

*associated with e*ach

state.

Que. Explain proceduse for transforming a Mealy machine com

into a Moore machine.

wa *26/8/13*

Dize

*Mealy Machin*e st

o s *Mealy machine is defined by 6 tuple.*

M=( Q, E, A, S, 1,90) where,

Q = finite set of internal state. Z = finite set of input alphabet. A = It represent the output alphabet. (5 ) f = Mapping Function no = Initial state. » It is a mapping from a to A giving the output

associated with each transition, ie input

Que. Constr*uct equivalent mealy machine f*rom a given moose

*machine*

**Transitio**n Ta*ble f*or Moore Machine &

z a b x to 9 a T

22

42

42

D*E 26/8/1*3

*Diagram for Moore Machine*

*(M.F.) for Mealy Machine*

+(90, a) = x( 8(90a))

X (93,a)= X(+(93, a))

X (93, a) =

(93, b) =

ol

($(931b))

|x (90, a)= il

(20, b) = 1(6(90, b))

= x(9) [x(90, b)= o)

x (91, a) = x(6(ai, a))

X (93; 6) = 0

Li (91, a)- o x(a, b) = x(6 (41, b))

x(91,6)= 1 102,0)=(80%,))

x(92, a) = 1 x (22, 6) = x(&(a, b))

(2) (92, b) = 1

HAKTI

*26/8/1*3

*Transition Table for Mealy Machine*

92 92

93

94

Diagram for M*ealy Machine*

gio, blo

91 92 91

23.

Diagram Fo*x Moor*e Machine 6+

DN*E 26/8/*13 178

*(M.F.) fox Mealy Machine* w a t

*\*(*q,a) = \*(869a))

*TT for Mealy Machin*e xx90, ) = ( 8(90, a))

ao a o q 1 q93 122 1

1

X490,0) = 0 x(90, b) = x(8 (20,6)

= X(22) x (90, 6) = 1

(91,0) = $*(&(*91,a))

= X(92) 1(a, a) = 1

*(*91; b) = \*(5(9,b))

= x(22)

92 | 22 93 ao

4 0 23 1

*Diagram for Mealy Mach*ine :

(92, a) = X($(92a))

= \(92) Li(d,d) = 1 x(Q2, b) = x (6(92, 6)).

= x (91) +(92, 6) = ol

(93, a) = x 78 (93, a)) nishably noonto

= \*(90) (93, a) = 1 x (93, 6) = 1 (8 (93,6)

= x (93) (93,b) =

SHANTI

SHANTIL 1931

DE 30/8/13

*\**

*Application of finite Automata*

*\**

\*v.v. ime

*State the application of finite Automata.*

*Soz*

*5*-08

OR

5-10

*state the role of lexical Analyser in F.R.*

→ There are v*ariety of software designed probl*em that are s*implified by automatic conversion of negul*ar expression notation to an ef*ficient computer implementati*on o*f th*e correspon *ding finite automata such application*s are

L Lex*ical An*alysen

*The token of programming languages are alm*ost witho*ut a regula*r set,

*A number of lexical analyses take a input o*s a *sequence of segular expression produced a single finite* a*utomata recognizing any token. They convert segul*ar *expression to an NFA with e transition* f *then to DER, Any*

*final state indicate a particular token found the transitio*n

*function of finite automata encoded in one of th*e severa*l* ways *to take less space. The lexical analyser encoded by th*e gener*atow is fixed program that represents coded table. T*his le*xical analyser may be used as module in compiler constructi*on

2. Text Editor

Certain tex*t editors are similar p*rogram the *substitution of string for any staing matching with given* r*egular expression,*

*SH*AKTI m*are:* 318)13

Tsee it A graph is called tree when it is connected & has no cycle There is always one special vestex called scot node & has no puedecese other than the soot node it has exactly one puedecessos

The node level 1 is soot node, Root is drawn at the top A sucessori s*co*t node is called son f the psedecessos is called pausen

*Son* of the same father is called sibling. The nodes not having any son as called terminal leaf. The depth of a tree is a node having maximum level.

Root node

I

loveli

Root of tree = node

nd

Height of tree = level3

thday, w

Two-way *Finite Aut*omata: \*

*S-*09

fuo

cted.

Que. De*fine two-way Finite Automata*

A two-way *deterministi*c *fini*te a*utomat*a (DER) is de*fine by 5 tuple.*

M =(4, z, d, 90,F) where,

Q = finite s*et of int*ernal state, E = *finite set of input alphabet.* 20 = *initial state.*

F = *finite set of final state* d *= Mapping function*

Si G8 E QX (L, R)

w *3/9/1*396

If $(2,0) = *(PL) it means that aRDFA in st*ate, *scanning input symbol a' the RDFR enter state '*f mov i*ts head left one squase.*

*similarly,*

*if* $14*,a) = (PR) then 2 DER in state q scanning* imput sym*bol a The DEA entess state '*p' & moves its *head sight to one square.*

In des*cription of the behaviour of one-wa*y finite automata we exten*ded* . fo @*xz this co*rre*spond* to thinking of the Finite automa*ta as receiving a symbol* en input channel process*i*ng *the symbol* f *sequesting the n*ext. This is insuffi cient for two-way f*inite automata sin*ce the two DEA may *move Left therefore we introduced an instantaneous description* (ID o*f 2 DER*. & *instanteneous description* descr*ib*ed the *in*put string c*urrent stat*e & *current position of the i*nput *head then we introduced the relation such that I, I*, if an o*nly if two way finite automata can go f*rom Il to 12 **in one moves**

*The ID war sepresent*s - 0 wx is i*nput string*

2 *q is current state.* @ *The input head scanning the 1*st sym*bol* of n

If w is accepted by n it starting in stute do with w on the input tape of *the head at t*he left head **of w & enter** *into the final stat*e.

SHAKTI

*v 4/9/13 9*9

*We can show that*,

Cb cca. so we *conclude that,*

ca=ch *Hense, our ass lemption that can b# ¢ is wsong*. *There fore we conclude that cancb=0*

*Warue. Define transition diagxdan*

Tra*nsition diagram is a diagramma*tic represe*ntation of states for a particular input* symbol.

90

a

92

4220

,

Transi*tion Diagram*

S7w.

\* transition ta*bl*e :

It is *tabular representati*on of transition *diagram es mapping function.*

Dr 4/9/13

loo

*Transition Diagram*

*Tud*os*ition Table*

\* Gontext Free Grammera

W-13

s-ot

*g*ue*. Define context f*ree Gr*ammes with suitable example*,

Gra*mmes is set of formal sules which check co*rrect *ness of th*e se*ntenc*e as cani co*nstruct sentence using this sules.* The se*t of sule denotes a set of valid sentence* fis*ich set of* valid sen*tence is called l*anguage.

A context free Gra*mmes i*s a *fini*te set of vasiables non terminal. Ea*ch of which rep*resent a language

A CF G is define by four tuple..

6 = (VITAP,s)

sowie where,

V = finite non-empty set of variable known as non termina T= finite set of tesminals. P= finite set of production

S = special vasiable known as START symbole for ext

s*aB/LA*

A alas/6AA *B b/bs/AB*B

N.T

V = {S, A, B }

T=34,6} P= S-BbR, R-aasbAR, B-> bbs Jabb S=2 snabb]

i 34/9/13 12

A p*roduction of the form A -> where,*

*A is non terminal.*

D

a to *f*ais terminal is known as co*ntext Free Gramme*s As we observ*e that in this type of grammer there is sestriction* that on the *left hand side of each p.)oduction th*ere is always non-ter*mina*l

A language genera*ted by CEO is th*e set of all sting of the termina*l that can provid*e from the start symbol's using the p2oduction & su*bstitution su*ch language is known as context fre*e languag*e (CEL).

\* Dazzvation Tage am *Parse Z*uzee \*

S-3

5-10

S-08

\*Se*que. Define Derivation Dree with suitable exampl*e.

*The derivation in context Free gramm*es c*an b*e Represented using tee s*uch tree rep*resenting derivations. We called des*ivation tree as pause t*ree.

A desivation tree *is defin*ed by the tuple.

4 = (1,T,R,S) whee

V = Einite non-empty set of variable\_known as

non-texminal. T = finite set of texminaln y p = finite set of production

. Especial variable known astust symbol

@ ©

There we two types of devotion tres

leftmost derivation tee. Rightmost desivation thee.

| SHAKTI

are: 2419|13

103

1. Leftmost Desivation Tees

In derivation wheneves we replaced any vaiable using a production there is no variable to the left of x(non te mind)

then it is called leftmost desivation tree.com

ition

all

2 Right*most Derivati*on Thee :

In the deriva*tion whenev*er we se*pla*ced any variable using a prod*uction & there is no variable to the right* of x (non terminal) *then it is called sightmost desi*vation tree,

3. A*mbigeous Gaamm*en \*

Sur A gra*mm*er is s*aid to b*e ambigeous it we can have more than one der*ivation f*or same stsing i.e. eithes both de le*ftlin*ed. D. *sightlined.*

L

OREN *Derivation far leftinear* B s*ightlineas are exactly* Same such grammes is *hooron os ambigcous gramm*es

can be is are

*Example 5 arala*

*Asbalas/bas* Construct leftmost of s*ightmost Date f*or stsing w*aabban*

let,

*A sbalasba leftmost Desivation*

S*-aAa* S-> OSAL S. ->d*abra*

(1=sha)

*DEL*AKTI

br 25/9|13 104

*WE* W

EST

s

(A=ba)

aabbad Tw=aabbaal

*leftmost Deivation Tree*

*Rightmost Desivat*ion

sa na

s → a*sboa (A=Sbr)* s → *asbbda (neba*) s → a*abbad* (s*a)* w=aabbad

*Rightmost Desivaton Tsee*

LDT *RADT De exactl*y same bence, that pastille

exammes is called ambigeous grommes

2*25/9*13

*Q*ue. Ex*plain ambiguity of CFC. It is a grammes*

*S -> sbs/o show that ce is ambigeou*s. - let.

$ => *sbs/a leftmost Derivation*

ssbs sabs - (s=a) S aba ( (sta)

w = abal

le*ftmost Derivation Tree*

*Rightmost Desivation*

s- sbs ssba saba (sa)

I wsaba

Rightmos*t Derivat*ion Tree

*LIDT PRD.are exactly same then above grammes i*s ambigeous

grammes

SHAKTI 07*:25 1*9 13 10€

*Qu*e. Cons*ides a grammes S QA*B

Ab Bb B ALE

w = abbble

o

7 *leftmost Derivation*

s -> *AAB* s abB*bB* is ab Eb B

S abba sz*abbbbb*

s abbbeb s abbbb

w=abbbb

(A=bBb) (B=E) (B=A) (n=*bBb)*

(BE)

*leftmost Derivation fre*e :

Right most Deriva*ti*on 4

s UAB -

s

S*- AA*

*abah* staabeh

*(B*=A) *(n*=bBb) (B=E).

-STATE

1 SHAKTI

Dare 25/9 13

saabb

s ab Bb bb s abbbb

w= abbbb

(A=bBb). (B= E) x x

Rightmos*t Desivati*on Tree

сах Astax

= 6

ist axbba

Oxaabb

bogor

1. s xbaa lax

x xalxble

w= aba abb *Jeftmost De*siva*tion*

S axbaa

s < xabaa (x=xa) s xbabad (Xexb) s >> xababad (x=xa) s z ababad (X= e)

leftmast. Derivation Tsee +

are 25/9/13

*Rightmast Desivation Tee*

LDT & RDI are exa*ctly sam*e then above grammes

*is ambigeous gramm*es.

1. E *production on null productions*

Que. Det*ine e-production with suitable example,* -

*A context free grammes may have puoduction of th*e foam A l is jus*t used to esased a so th*e production of the Foxm Ane where A is n*oor teminal is known as puoducti*on O*m null production*.

- A production in a CFG is nullable if

it sepsesent en Osmos*e noi of substi*tution.

e*xample* →

S > AB A ARA/e B BBB/E

Att

7

given

SHAKTI DE: 2619/13

avi

s — AB/A/B A - q*aplanlarl*a I A CAR aalal

B -> *BB/b*elbelb 1B -> Belbelb]

CEG wi*thout e producti*on -

ISAB/A/B

2-*dadal*a B B B B /bBlb

S - ABAC A > *OR/*E Bbble

له خه و

given

S S

ABAC | BAL/AAC/ABC/AC/AC BCC ABAC BAC DAC) ABC ABCLC

A > en a B -> b Blo cd

CEG without production -

ad . - ABAC BACAAC ABC AC BCC A unla B. bolb

SHARTI

on 27/9/13

16

P*e Unit* produc*tion*

*Que. Define unit production with suitable.example*. -

*Any production of context free gidmhet is* ofithe *fo*rm 1 -> whereafB is non *terminal is known as uni*t product*ion*

ex

Remove unit pro*duction from th*e following CF.

SAD

B Al bb

↑ Aalbol Step 1 - It doesk*ortai*n production

lower to

step 2 - *It contain unit producti*on 50, semove it.

Azalbe/B A *It contain unit production s* sosemove *B b*y

psed*uction (B+b7* Aalbclbbl

uppes scan \*smole non temind

wemove it

B > Albb It c*ontain wit paduction t* o rem*ove a b*y

production [At albeibbl. B *a|bc*|bb|bb

B albel bb S ALIB It contain unit paoduction 'B' sosemove B by ibubapkoduction B abel bb

s nalalbelbol

SHAKI

mur: 2719)13

GFG without unit production -

s Aulalbel bb B al bel bb

A albclbb

VR

Remove a*ll unit p*soduction from the CFG.

s Albl

A Blb.

1 B - Cla → step 1 - It do*esítcontain* e pwoduction

step 2 - It contain unit poductions, Bemove it!

Bsla It contain wit poduction's so semove s by Lolci spro*ducti*on Lisbb]

B. bblaul, IoT

BO

T

A Blb It con*tain unit production 'B*'soemove B by

TI *paeductio*n (B2bblal Tabbladet

s > Albb It con*tain unit production't* sosemove by

production [ bbla/b] s-*bbb*bulb