Turing Machines

160

Theory of Computation

161

DETERMINISTIC CONTEXT FREE LANGUAGES (DCFL)

& Show th*at t*he CSL's are closed under the followin**g operation:**

1) *Union, (ii) Intersec*tion. (iii) Concatenation, (iv) Substitution. Design a turing machine to recognize the set of strings with an equal number of O's and 1's. **Define multihead turing machine and non-deterministic turing machine. Construct left-linear and right-linear grammar for the language**

o 1 (0+1)) 12. Prove that CSL's are closed u**nder concatenation and intersection,** 13. Design a turing machine to recognize th**e set of strings with an equal**

Sumber of O's and I's.

Show that CSL's are closed under:

Substitution and (ii) Inv**erse nomomorphism.** $ Explain:

c) Off-line turing machines (ii**) Deterministic two-stock machine.** 16. Construct left-linear and **right-linear grammars for**

On+10)\*11\*00) 2. Design turing machine to recogniz**e the following language**

10" "0" n> 1) Show that it is undecidable whether a T**M halts on all inputs.** Prove that if L is accepted by a non-deterministic Turing machine M. then L is accepted by deterministic Turing machine M,. Prove that if L is a regular set, then L is generated by some left linear grammar and by some right linear grammar.

The class of languages.accepted by push down automata is *conte*xt *free languag*es. There are two types of PDAs namely:

(1) Deterministic push down Automata. **(2) Non-deterministic push down Automata.**

The deterministic PDA accepts family of languages called Deterministic Context Free Languages (DCFL's)

When a compiler designer wants to use use compiler writing systems then he should choose a syntax for his language that makes it a DCFL. The designer should be able to determine whether the proposed language is DCFL. If it he can prove it by constructing DPDA or LR-grammer for it. The LR-grammer have the property that the grammer generates cxactly the DCFL's.

000

3. Normal Forms e*f* DPDA Definition: Let M=(Q, EI'8. 9o. Zo. F) be a pushdown automaton. Mis said to be deterministic. ifit satisifies following conditions. (1) if Siq, a. X) has atmost one element for any qe Q. a in !

and Xer (2) if 8(9, €.X)\*0. for any qe Q and Xelthen Stq.a. X) = 0.

for every ae E. **A language Lis sa**id to be DCFL iff there is a DPDA such that l=L. M!

Deterministic Context Free Languages (DCFL)

162

Theory of C**omputation**

- Lemma 1 : Every DCFL is L(M) for a DPDA. M = (Q. . T. 8, 2,. F) such that if & q, a. X) = (p. Y) and

Proof: Consider 84.a. X) = (r. Y) and Y>2.

Y = Y,Y ---Y with n23 Create new non-accepting state Pie Pan Pir- and redefine.

di4. a. X) =(, Y-Y.

Sip. e. Y-) = (Pil. Yn-I-YO) for all I sisn-3, and

(P.-2. E. Y.)=(r. Y, Y,) Now (q. a, X) (p. e. Yo-Y)

tip. €, Ya-2Y - Yn)

t (P. E. Y-3 Yaz Yn-IY)

(P.-2. E. Y,Y, ---Y)

Fire. Y, Y. --- Y). That means. in state q, for input a, with X on top of the stack, the revised DPDA still replaces X by Y = Y, Y, --- Y*,* and enters state r but it takes in-I) moves to do so. This is contradiction. Thus, the DPDA can never push more than one symbol per move.

• Lemma 2: Every DCFL is L(M) for a DPDA.

=IQ. L. .8.4.. Z.). then Y is either Ela pop). X (no stack move), or of the form YX (a push) for some stack symbol Y. Proof: Consider L=L(M"). where M' =(Q.L.T. S. 46. X, F') satisfies lemma l Construct M simulate M' while keeping the to**p stack symbol of Min** Mi control Fomally. let

Q=Q'xr. 4o =196. X:l. F=F'xr' and rur U IZ! where Z. is a new syimbol not in

Now 8 is defined by (i) 8([q, X]. a. Y)=(p. Y). e) for all Y,

if S'q. a. X) =(p. E)

(That means, if M' pops its stack, M also pops its stack). (ii) ([q. Xl.. Z) =(p. Y). Z) for all Z,

if 8'(q. a, X)=(p. Y) (That means, if M'changes its top stack symbol, then M records

the change in its own control, but does not change its stack) (i) ([q. X]. a, W)=([p. Y]. zw) for all W,

if &'(q, a, X) = (p. YZ) (If stack of M' grows, M pushes a symbol onto its stack) .. By induction on the number of moves made (4o. W.Xo) (q. e. X,X, ---X.)

if and only if (196, X,]. W.2) ([q, X,1. c. X,X, ----XZ,)

Thus L(M)=L(M). . Lemma 3: Let M be a DPDA. There exists an equivalent **DPDAM' such that on every input, M' scans the entire input.** Proof: Let M=(Q. 2. I, 6, 40, F). Define M'=(QU 196. d, f). . T U X), '. 46. XOFU () d is a dead state where no next move is possible. For any input symbol. the only transition from d to d occurs, but no change of stack occurs. 8 is defined as: (1) S(q. E. X)=(qu. Z. X.): X, marks bottom of stack. M IS(q. a, Z) = 0.89, E, Z) = 0 . for some qe Q. a € L. Zer

then &[q, a, Z) = (d, Z). Also, for all, 9 e Q. a EI.

8°14. a. X.)=(d. Xg).

Seterministic Context Free Languages (DCFL)

164

Theory of Computation

Tbeorem, Tke DCFL's are not closed under:

a Union

by Concatenation **(c) Kleene closure** (d) Homemorphism. (a) For union : Consider languages.

L, = 01211. 20)

Ly = (0' | 2 | i, j2 0} Now, we have to show that L, and L. are DCFL not closed under union.

L=L, UL,

= 0' 1' 2 |. i20 U 10' 1' 2|i. 120 }

L = 10' P' 2 li=1 = 1 But from example 1. L is not DCFL.

.. LEL, UL is no: DCFL. - DCFL's are not closed under union.

(1) 8 d. a. Z) = *(d.Z). f*or all ae2, ZETU [X]

*If (*q. ,Z) (q. €. Y) then

S'q. e Z)=(d. Z) provided no q, is final and *st*a. c. 2)=*(f*, z) whenever one or more of the qs is final.

814. s. Z) = (d. 2) for all Zin TU [X] vi) For any qeQ. DEZU (C), ZET, if &'(g, a, Z)

has not been defined by (ii) or (iv). then define

(q, a, Z)=8(q, a, Z) Then (M)=L(M'). M uses all its input. Suppose for some proper prefix x of xy, then we have an ID of the form

19. y. Z., Z2Z X,), such that 19h xy, Xo) M (9. y. Z, Z, ---Z Xo) By (1) it is not possible that M'halts. By (iv) it is not possible that M' ma**kes an infinite sequence of e**

moves without erasing Z.

M' erases eventually enter an ID (q'. y. X.). By (ii). (q. %.X) (d. y. X.): where y = ay'. a 2 Hence M' reads all the inputs. The language L=1012kli=j or j=k) is not DCFL. Let L be generated by a gram**mer whose productions are**

s - AB CD. A 041 € B 2B e, C OCE

D 102 € sing Odgen's lemma, we can prove

L O 12 lixj.jk) is not a CFL, hence not a DCFL. Also L=100 12" As L is not DCFL. I is not DCFL. Hence L is not DCFL.

MP

(b) For concatenation:

Consider language. La = L, UL. = L, is DCFL, because the presence of symbol 'a' tells fora word in L, or absence of symbol 'a' tells for a word in L, sa is a DCFL But a' L, is not a DCFL. Then L. = 1 L n a o'12\* would be a DCFL (As DCFL's are closed under intersection). But Li = al, UL If L, is a DCFL. accepted by DPDA M. the.. we could recognize LUL by simulating Montimaginary input and then on realmur As L UL is not DCF1.

L is also not DCFL. ... DCFL's are not closed under concatenation

**Deterministic Context** Free Languages (DCFL

167

166

Theory of C*omputatio*n

**e Por kleene closure:**

Consider language, L = (a) UL 2 l, is a DCFL.

i is not a DCFL (As DCFL are not closed under concatenation)

DCFL's are not closed und**er kleene closure.** id) For homomorphism:

Consider language, La = aL, UbL Let L is a DCFL. Let L be a homomorphism that maps b to a and maps other symbols to themselves.

hi L.) = L; But L, is not DCFL

> L) is nor DCFL.

DCFL's are not closed under homomorphism. Theorem: Let L be a DCFL and Ra regular set. T**he following** problems are decidable

(1) Is L=R? (2) Is RSL? (3) Is L=f? (4) Is L a CFL?

(5) Is L regular ?

Proof: () Is L=R! Let L be a DCFL and R is a regular set. 1 L = (LOR) U (

TR) = 0 As DCFL's *are eff*ectively closed *und*er complementation and intersection with a regular sei R. Also, the CFL's are effectively closed under union.

= L; = 0 is a CFL. Hence, emptiness of CFL's is decidable. ... LER is decidable. (2) Is RSL? Let L be a DCFL and R is a regular set. Consider InR=0 As LOR is a CFL. Hence, emptiness of CFL's is decidable. = LOR=0 is decidable. . ESL is decidable. (3) Is I=f*?* Let L be a DCFL. As the DCFL's are effectively cl**osed under complementation.** . [ is a DCFL. .. L=0 is decidable. (4) Is La CFL ? Let L be a DCFL. . I is a DCFL . I is a CFL is trivial for DCFL's. .: I is a CFL is decidable.

(5) Is L regular ? Let L be a DCFL. Frem (1). Regular sets for DCFL's is decidable. 2. L is regular and decidable.

Dela Crest True Languages (DCFL)

168

Theory of Computation

Construct the menempty sets of items **for the following grammars** *hi*dare LRO

T; -+1 E,

T, -+ Ey

T+ (E,

T; -

E; +

Predicting Machines

For a number of other closure properties of DCFL's. there is need for construction in which the stock symbols of DPDA Mare modified to patitain information about a certain fine storiton A. The information

*so*ciated with the top stuck symbol indicates that for each state of M and pof A. whether there is to input string that causes Mo on when started in state with its current ack and simultaneously c es to accept if started in step

ETE,

be a mommal form DPDA and

A1Q, I s ou Full The predicting machine Mand A in defined as

ht. Alur A, 8.

45

IT MAX is in Dr. A. IZ. HY). then consists of exactly there paits (q.p) such that there wei' for which Alp. w is in F and (q. w. ZR (S. a) for some Se FM and e. Bet witere B is the string of first component of .

168

Theory of Computation

Construct the enempty sets of i**tems for the following gratinars** wachane LRC)

E

T,E,IT:

TjE

& TE, E, -+ TE:

Tj

-

g +

2 Predicting Machines

For a number of the closure properties of DCFL', there is a need for construction in which the back wymbols of DPDAM modified to contain information about a certain femme ato AThe information

sociated with the top stack symbol indicates that, loreach uleg of and pot ether there is one imput string that ca M acet when started in te with its current stack and simultaneously to accept it started in

TE

be a small form DITA

Then predicting machine Mand A is defined

MAQ, LT. X Ful where is the of the of If HM. A) is in ID (CA. 12. wy). then ja.consists of exactly the pairs 14. p) such that there is no I' for which p int und 19. w.ZD S for some S ander where is the string of first component of T

170

Theory of Cemputation

Deterministic Context Free Langunges (DCFL)

B aSb Ba. Sb

B - asb B - Sb.

7.3 LR (O) and LR (K) Grammar

Before studying LR (O) grammeritis mandatory to learn basic terminology needed to understand the concept of LR (O) grammar. LR-Items

1) Item: An item for a given CFG is a production with a dor where in the right side inlcuding the begining or end. In the case of an È production

B e . B . is an item, (2) Handle: A handle of a right-sentential - from y for CFG. is a substring B such that

SAW = sow and

BBW = In other words a handle of Y is a substring that could be introduced at the last step in the right most derivation of y.

(3) Viable Prefix: A viable prefix of a right sentential formy is my prefix of y ending n*o f*arther right than the right end of the handle of 7.1. Anght-sentential form is that can be derived by rightmost derivation).

(4) Valid Item: An item is said to be valid for a given valid prefix, of *the do*t is the nightmost symbol in the item.

Obtain the items for the following CIG

S + AC

A - BSA

BaSb abc The items for the above grammar are

S +. Ac $ - A. s → Ac -

B - ab

B - ab BC

в - с. Consider the grammar G=(V, T, P, S) where P= {S' - Se, ASA, A - aSb | ab). **Find the handle for the right sentential form Sa**SbC.

The right sentential is obtained as follows:

S' SC

SAC

- SaSbc This SaSbc is the right sentential form and aSb is a handle.

**Find the viable prefixes for Sa**Sbc. The viable prefixes are c, S. Sa, SaS and SaSb.

Star @

SS SS-500C 5-Sco

A

$-A A S-AU

Aasba

ហ ថា

-SbA

->ab)

. BS

Bos A → BS.

S-SA 4-Seb Arab

Fig. 7.1: NFA recognizing viable prefix.

Deterministic Context Free Languages (DCFL)

*1*72

Theory of Computation

Computing Sets of Valid Items

*The* knowledge of valid items for each viable prefix Y is essential for better understanding of LR(O) grammers and L(G) generating LR O). It turns out that for every CFG, G what so ever the set of viable prefices is a regular set, and this regular set is accepted by on NFA whose states are the item for G.

Applying the subset construction to this NFA yields a DFA whose states is response to the viable prefix Y is a set o**f valid item for Y.**

S-SC SSC

(

SS.C) Start S SA

1SS.A S

A- Sbi AUS

A- ab Aab

Let M=(Q. V. U.T. 8. (v.), where is a ser of items for Plus the state q which is not item Deline: (1) 8 (4. 6) = (

S a S a is a production +8) SEA -BBC) (B- 1B - od poduction!

tu (A - XB. XIA - OX.B! O Consider the grammar

G=(IS'. S. A). (a, b, c). P.S) where P = (S' SCS - SALA. A - asbab Construct an NFA for re*co*gnizing viable prefixes for the above pranimer:

The NFA for the above grammer is shown in the following figur*e*

(s A 181 S*ta*rt A-BA ($)

A 181 B B fa. b. $)

B-bla. b. $)

A-

ISSA 1S .A AwaSb A teab

(A-Seb)

3-Sea AnSb A-teab

(B-Babs B A*B. (a*. b. Si

A BAS) ABA ISI

B

B

b

fa. b.$}

ab.. fab.

A

.

-h

Fig. 7.2: DFA where states ar**e the sell of valid items.** We cannot construct an NFA for accepting the viable prefixes for GWIN. T. P. S) follows.

fa. b. $]

Fig. 7.3

171

Deterministic Context Free Languages (DCFL)

170

Theory of Computat*i*on

B + Sb B → a. Sb B --> AS b B - aSb. B . ab

73 LR (O) and LR (K) Grammar

Before studying LR (O) grammeritis mandatory to learn basic terminology needed to understand the concept of LR (O) grammar. LR - Items

(1) Item: An item for a given CFG is a production with a dor where in thenght side inlcuding the begining or end. In the case of an production

B E . B . is an item. (2) Handle: A handle of a right-sentential - from y for CFG Gisa substnng B such that

S AW 8BW and

BW = In other words a handle of yis a substring that could be introduced at the last step in the right most derivation ofy.

(3) Viable Prefix: A viable prefix of a right sentential form y is y prefix of y ending no farther right than the right end of the handle of Y+ Aright-sentential form is that can be derived by rightmos derivation).

(4) Valid Item: An item is said to be valid for a given valid prefix, if the dot is the rightmost symbol in the item. o Obtain the items for the following CFG

SAC

A - BSA

Basb | abc The items for the above grammar are

S - AC

S - A.. S AC. 4 . BS

Bos ABS

B - ab в – с

в — с. C*o*nsider the grammar G=(V, T, P, S) where P ={S' Sc, ASTA, A → Sbab). **Find the handle for the right sentential form SaSbC.**

**The right sentential is obtained as follows:**

s' - SC

→ SAC

→ SaSbc This SaSbc is the right sentential form and aSb is a handle.

Find the viable prefixes for SaSbc. The viable prefixes are c, S. Sa, SaS and SaSb.

Start-@

$=$0 SS-500C-SCO

C5-SA EŠ>A 4-S-AD

eab

A-a-SDA- ab S-SA. A-Sob A-rabo

Fig. 7.1: NFA recognizing viable prefix.

Theory of Co*mp*utation

Deterministic Context Free Languages (DCFL)

172 Co*mputi*ng Sets of Valid Items

*The kn*owledge of valid items for each viable prefix Y is essential for better understanding of LR(O) grammers and L(G) generating LRO). It turns out that for every CFG. G what so ever the set of viable prefrces is a regular set, and this regular set is accepted by on NFA whose states are the item for G.

Applying the subset construction to this NFA yields a DFA whose states is response to the viable prefix Y is a set of valid item for Y.

/S-SC. SSC

(sys.cl S-.SA

SS-A S .A

A asbl AnaSb

Aab Aab

Let M=(Q. V. U.T, 6.9..0). where is a set of items for G. Plus the state, which is not item. Define : (i) 8140. E)= 15 → also is a production!

) SIA => O-BB ) = (

B B y production! ti SIA - XB. XI=1A aX BI Consider the grammar

G=({S, S. A). (a, b, c).P. S') where P ={S SC S SATA.A - asb abi Construct an NFA for recognizing viable prefixes for the above grammer.

The NFA for the above grimmer is shown in the following figure

40

- SA

9. (5 AISI JA BAS)

A .151 B aBla, b, $1 B bla. b.$}

A .SE Ab

S-+SA 1st.A AtaSb Aaab

AS b

3-S. AaSb A+aab

S-SAS

(B- Bja. b. Si

1. B B AB. (a, b, STB

B- bla. b. Si

ub

(A → B-AISI A BA (S) AISI B aB a. b. Si Bh. lab.si

Aabe

-

(5 th , file hos)

Fig. 7.2 : DFA where states are the sell of valid items.

We cannot construct an NFA for accepting the viable prefixes for EGE G=(. T. P. S) follows.

( ABA. 141

Fig. 7.3

175

Deterministic Context Free Languages (DCFL)

Delinition of LR (0) Grammar

4 CFG G is and to be LR (O) grommur, ir, Hlis starts symbol does not appear on the right side of any luction, and

1) Forevery viable prefix Y of whenever A-ya is a complete item valid for Y then no other complete item not any item with a terminal in the nght of the dot is valid for y.

Show that the grammar G = ({S, S, A). (a, b, c), P, S) is LR(O) - Constructing DFA from NFA recognizing va**ble prefixes for grammer** G. Fig (2) in Ex. (3). It is clear from the figure that of these states, all but 1.1..1, and I, consist of a single item (i.e. states L, I, II, and I). Moreover, the states with more than one item have no complete items, and surely S the start symbol does not appear on the right side of any production. Hence grammer is OR (O). Definition of LR(1) Grammar

LR (1) stands for "left to right" scan of the input producing a right most derivation and using 0 symbol of lock ahead on the input.

A CGF G is said to be LR(I) grammer, if

the start symbol appears on the right side. ) whenever the set of times I valid for some viable prefix inlcudes

some complete item A a-ainea,1 then (1) No a appears immediately to the right of the dot in any item

of land (2) If B-B. (b, b2, bz-a is another complete item in I,

then a, b, for any lsisn and 1sjsk

Theory of Comp*ut*ation

Let us consider a DFA for set of item for G. This DFA is show in Fig. (3) in Ex. (4). From figure, it is clear that the sets of item 1.1.1, and I, consist of only one item. Set I, has one complete item, A+..(5), but but does not appear to the right of the dot in any item of I, A similar remark applies to I, and I h**as no complete item.** Thus, gra*mm*er G is LR (1).

Ionut Definition of LR (K) Grammar

LR(K) stands for "left to right" scan of the input producing anght most derivation and using K symbol of lock ahead on the input.

Eroperties of LR(K) grammar, (1) Every LR(K) grammar Gis unambiguous. (2) If G is an LR(K) grammar then there exits a deterministic

push down automata (DPDA) A accepting L(G). (3) If Ais DPDA A, there exits on LR(1) grammar G such that

L(G) = N(A). (4) If Gis on LR(K) grammar, where K>1, then there exits an

equivalent grammar G, which is LR(1). In so for as languages are concerned, it is enough to study the languages generated

by LR (O) grammar and LR(1) grammar. (5) The class of deterministic language is a proper subclan of the

class of CFL. (6) The class of deterministic CFL to closed under

complementation but not under union and intersection. (7) ACFG is generated by an LR(O) grammar iff it is accepted

by a DPDA whic**h has prefix property.** (8) There is an algorithm to decide whether the given CFGS

LR (K) for the given natural number K.

0 Consider the grammar G

S-A, A-BA €, BaBb Show that the grammer G is LR (1).

**Deterministic** Context Free Languages (DCFL)

1 theory of Computation

177

A The Chomsky Hierarchy

Where aand ß can be strings of T and NTs. Such rules would

rmit arbitrary substitution of string during derivation as reducuon hence Grammers are classified into four categories based on the nature (av are not

Tassed to four cilesores Dased on the alle ey are not relevant to specification of programmi*ng* langu*ages. Thes*e productions used in them (Chomsky 1963). These categories havammar are also called as unr*estric*ted grammer following are some characteristics or associated practical implications which make then molamples of unrestricted grammar or less attractive for programming language specification. The fou (1) Let L be the set of all palindromes over 10.1). then the **categories of grammer are: type-0 grammar** (Unrestricted grair merrestricted grammar G generating Lis **type - 1 grammar** (Context sensitive grammer), type-2 gramma

G=(1S), (0. 11. P. S) **Context free grammer) and type-3 grammar** (Regular grammer). Where

Where Pis S - OSOISIO SIC o Type-O grammar have relations with the languages accepted

by turing machine.

(2) Let L=WoWR WE(0D). then the unrestricted

m**ar generating Lis . Type-1 grammar are relate**d to Linear Bounded Automata

• Type-2 grammar have relations with Push Down Automata.

G=({S). 10. 1), P.S)

Where Piss - 0501011C

**• Type-3 or regular grammar are related to Finite Automatas.** This relationship is shown in following diagram,

(3) Let L = 'WWXW e*1*0.1). then the unrestricted grammar

enerating Lis Type

Turning Machine

G=([S). 10. 1). P. S) Contex sensitive

Where Pis

Linear Bounded or Type-I

Automata

S - ABC. AB - ADAE.DC → BaC. EC Bbc. Contex free

Push Down

Da + aD. Db 6D. Ea a aB + Ba. bB - Bb. grammaror Type-2

Automata

AB - E.

C E Regular

(4) Let L = lali is the positive power of 2). then unrestric

Finite grammar

mmar Gis or Type-3)

Automata

S - ACB. Ca C. CBDB. CB + E

aD Da, AD - AC. GE → E. AE - E te: In this type of grammar there is no restriction are defining the

luctions Fig.7.4: Languages **and its corresponding Automata.** Type-1 Grammars

These grammars are known as contrxt sensitive grammars because Type-0 Grammars

luctions in these grammars specify that derivation/reduction of **Th a mmar known as phrase slature grammars. contain rulecular strings cantake place only in particular context** of the farm

A type-I producdon has the form 0 :: = B

CAB : : = B .

Theory of Computation

Deterministie Contest Frer Languages (CFL)

Type-2 Grammars ma sentential form can be replaced by A Cory These animarsimpose no context requirem**ents ordenado** o ly when it is enclosed by the strings and B. These renme-typical type-2 production of the form

purticarly relevant tortogram ng language specificaties

al programming language elemente variables which can be applied independent of its context. These gramme

elit sensitive to the context. There is a restriction is therforektown as context free grammar (CFO). These gramma N ormal that in the productions

must be atleast a. b ase. In order to convert all productions into productions far contra

are ideally suined for programming language specification, gummers uma campesite symbols from the previous pratimar Type-3 Grimmars widt e the productions in the form of type gratuur,

Production rules of the form The complete set of composite symbols for the grammarae

A :: = Bt TAB]- Aal. IACal. (ADA) - JAEal. Cal. [Da]. [Ea). [2CB

B. (aDBI. JOE. ID.B) and ael I nice the resultar productions for the Suntext sensitive grammarare

Characterise type-3 grammars. Note that these rules also satisfy

S-LACaB]

the requirements of type-2 grammars. The peculiar farm of the RHS

alternatives namely a single terminal symbol or a terminal and a non

16]laBlaCaB).

te*rm*inal symbol give some practical advantage in scanning. However, the

TAC

nature of the production rules also make such graminars rather restrictive. -+ [Aa) a [Ca].

*Hence,* such rules are generally used for the specification of texical 1ACa] [AB] - [A]a (CaB).

components of a language. Viz, variable names, labes etc. TACIB] - [A] [«CB].

Type-3 grammars are also known as linear or regular grammars, CoB] - [GCB)

**further categan**ized into left linear and right linear grammars depending on 3) (GCB] [ACB!

**whether the** NT in the RHS alternative appear**s on its extreme left or**

4) [BCB] [BE

extreme right.

Daj Dala.

If all the productions of CFG are of the form A W B or A LaDB] [DB]

W where A and B are variables and W is a set of terminals (possibly 1Aal Dal -[ADala.

empty) then we say that the grammar is right linear. a [DaB|--[DalaB].

all the production of CFG are of the form A BW or A-W (Aal(DaB) (ADB)

where A and B are variables and W is a set of terminals (possibly empty). 161 1A Dal ACA)

then we say that the grammar is left linear.

TaEJ - [E] LASTE- JAEJ

Deterministic Context Free Languages (DCFL)

180

Theory of Computation

O Construct left linear *and r*ight linea**r grammer for the following** language

10+11 00 (0+1) Left linn

Right linear - Ssi LADO

S OSTISIDA

A - OATIA where 0 =(15. Al. 10. 11. P. 5)

P is as above.

- Left linear

Right linear S - S105111A

SOSA A - Adle

A-10ATIAE white Glis. Al. (0.1). PS) P is as above

OR SADAIB

SOSIALE ASILI

A - OB IB01 11 - 130

BIA where G = {S. A B1. 10. 11. P. S)

P is as above con-1011\*00) Left Linear

Right linear SA001

SAIOA | OLAE A + B15

A-10AOIA IIB TAB 1- B110A

B-Oos dere GESABU. 10. II. P. S)

P is as above,

7.5-Closure Properties of Families of Languages

Note: We make a general approch and study all families on king having certain closure properties. This will pron de the new insight the underlying structure of closure prunernes and will simplif the studs of new classes of language.

Trios and Full Trios Family of Languages

A family of languages ist collection of languages u unny atkan one non-empty language.

Trios: A trois family of languages closed under intersection with regular set, inverse hamomorphism and E-free farvard homomorphism Since. *homomorphism his e-free mie for anymbol a.*

Full Trios: If family of languages is closed under all homomorphisms and well as inverse homomorphism and intersection with regular sets. then it is said to be full trio. It is denoted by

Lemina: Every full trio contains all regular sets.every trio contains all e-free regular sets. Proof: Case - 1 : Let jo be a trio.

I an alphabet and

RCI an E-free regular sets. Since, contains at least one empty language, Let L be in jo

SAW be in L Define I'= (a'la is in E) und h be the homomorphism that maps each o in Ito e and each 'Slow Then L = h- in because w is trio. As Wisin L.

..............

.

Detecmimistic Context Free Languages (DCFL)

Construet left linear and right linear grammer for the following

180

Theory of Computation

Left linear S - S0 S 400

4 - ADAI where G=(S. A), (0.1)..S)

P is as above

Right linear S - OSI IS00A A OAIA

Left linear S - SIOSHA

A- A06 where G=(S.A.10.1).P.S)

Pis as above

Right linear S - OS A A-10A IAS

7.5-Closure Proper

Note: We take a genu having certain closure prop the underlying structure of of new classes of language

Trios and Full Tr Family of Languages

A family of languages one non-empty language.

Trios: Atrio is family u regular set, inverse h homomorphism. Since.al any symbola,

Full Tree

5 - AD ALB A - SIB B - BO

OR

SOSIA

and