Decidiability :

some Questions always comes in our mind about CFG. Let us discuss some of them.

(I) How can be decide whether or not two different CFG define the same language. ?

(II) Given a Particular CFG, How can be decide whether or not it is ambiguous. ?

(III) Given Particular CFG, which is ambiguous, can we whether or not there is a different CFG that generate the same language but is not ambiguous ?

(IV) How can we tell whether or not the complement of a given CFL is also context free. ?

(V) Given a CFG, How can we tell whether or not there are any strings that is does not generate ?

Some other fundamental questions about CFG that we can answer:

(a) Given a CFG, can G tell whether or not it generates any string at all ? This is the question of emptiness.

(b) Given a CFG, we can tell whether or not the language it generate is finite or infinite. This is the question of Finiteness

(c) Given a CFG and a particular string w. Can we tell whether or not w can be generated by CFG.? This is the question of membership.

(1) Emptiness:

check emptiness:

step

(i) if CFG grammar have $A \rightarrow \alpha$ type production then put α in the place of A in RHS of all production where α is the string of terminals.

(ii) Repeat step (i) until it eliminates S or it eliminates no new variable. if S has been eliminated then CFG produce atleast one word (string), if not then not.

ex.

$S \rightarrow XY$
$X \rightarrow XX / AX$
$X \rightarrow AA$
$A \rightarrow a$
$Y \rightarrow BY / BB$
$B \rightarrow b$

so Replace all A by a in RHS of productions

$S \rightarrow XY$
$X \rightarrow XX / aX / aa$
$A \rightarrow a$
$Y \rightarrow BY / BB$
$B \rightarrow b$

Now B replace by b

$S \rightarrow XY$
$X \rightarrow XX / aX / aa$
$A \rightarrow a$
$Y \rightarrow bY / bb$
$B \rightarrow b$

Now X Replace aa & Y Replace bb.

then

$S \rightarrow aabb$

So S eliminates with atleast aabb. So grammar is not empty.

a CHECK Emptyness

$S \to XY$
$X \to AX$
$Y \to BY \mid BB$
$A \to a$
$B \to b$

Replace all A's by a & all B's by b

$S \to XY$
$X \to aX$
$Y \to bY \mid bb$
$A \to a$
$B \to b$

Replace all Y's by bb

$S \to Xbb$
$X \to aX$
$Y \to bbb \mid bb$
$A \to a$
$B \to b$

variable X in the Production $S \to Xbb$ will never terminate. so grammar generates no string. so it is empty.

Finiteness: A Language L generate from the CFG is finite if CFG is in CNF form and there are no cycles in the directed graph, generated from the production rules of the CNF form of CFG.

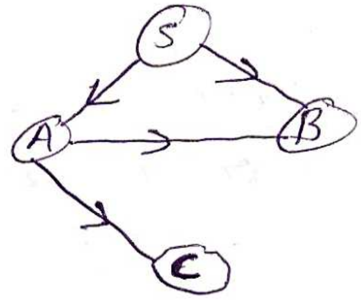
The longest string generated by the grammar is determined by the derivation from the start symbol.

No. of vertices of directed graph is the same as the No. of variables in CNF form of CFG.

ex.

$S \to AB$
$A \to BC$
$B \to a$
$C \to a$

The given grammar is in CNF form. So the directed graph is



the graph does not contain any loop. So the language generated by CFG is Finite.

Q. check Finiteness:

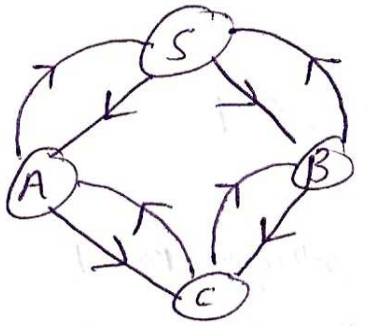
$S \to AB$
$A \to B$
$B \to SC / a$
$C \to AB / b$

The grammar is not in CNF so firstly convert it in CNF.

Remove unit Production

$S \rightarrow AB$
$A \rightarrow SC / a$
$B \rightarrow SC / a$
$C \rightarrow AB / b$

it is in CNF form so directed graph is



the graph contain loops. so the language generated by CFG is infinite.

membership Problem:

membership Problem decides whether a string w is generated by a given CFG or not. This is Proofed by CYK algorithm. according to this algorithm, the string of terminals is generated from length 1 to length of w. if w belongs the set of string generated then w is the member of the strings generated by the CFG. for this, we need to convert the given grammar into CNF.

ex.
$S \rightarrow bA / aB$
$A \rightarrow bD / aS / a$
$B \rightarrow aE / bS / a$
$D \rightarrow AA$
$E \rightarrow BB$

check that string "aba" "baba" is the member of the Language or not.

First we convert this CFG to CNF.

$S \to FA / GB$
$A \to FD / GS / a$
$B \to GE / FS / a$
$D \to AA$
$E \to BB$
$F \to b$
$G \to a$

Now produce strings of length 1

string	Producing variable
a	A
a	B
a	G
b	F

Produce strings of length 2

String	Producing variable.
ba	S ($S \to FA$)
aa	S ($S \to GB$)
aa	D ($D \to AA$)
aa	E ($E \to BB$)

Production is type of $A \to BC$ so 1 alphabet generate from B & 1 from C

Produce string of length 3 :
here Production type is $A \to BC$ so one variable generate two alphabet & another variable generate one alphabet.

String	Producing Variable.
baa	A ($A \to FD$)
aba	A ($A \to GS$)
aaa	A ($A \to GS$)
aaa	B ($B \to GE$)
bba	B ($B \to FS$)
baa	B ($B \to FS$)

Produce string of length 4:

in this part there are three posibilities.

1st variable generate 1 alphabet & 2nd variable 3 alphabet

1st ———— 2 " " " " 2 ————

1st ———— 3 ———— " 1 ————

string	Producing variable.
abaa	D (D → AA, A → a, A → FD)
baaa	D (D → AA, A → FD, A → a)
aaba	D (D → AA, A → a, A → GS)
abaa	D (D → AA, A → GS, A → a)
aaaa	D (D → AA, A → GS, A → a)
aaaa	D (D → AA, A → a, A → GS)
aaaa	E (E → BB)
abba	E (E → BB)
bbaa	E (E → BB)
abaa	E (E → BB)
baaa	E (E → BB)
aaaa	S (S → GB)
abba	S (S → GB)
abaa	S (S → GB
bbaa	S (S → FA)
baba	S (S → FA)
baaa	S (S → GA)

string baba generate from S so it is a member of its language. but in length 3 string aba generates from A not S so it is not a member of its language.

Pumping lemma for context free languages :

Let L be a CFG. then we can find a natural no. n such that

(i) Select an arbitary string z from L such that $|z| \geq n$

(ii) Let u, v, w, x, y are the substring of z such that
$$z = uvwxy$$
where $|vx| \geq 1$
$|vwx| \leq n$

(iii) if the given language is CFL, there must some v and x such that
$$u(v)^i w(x)^i y \in L \quad \forall i \geq 0$$

Application of the pumping leema for CFL :

Pumping leema is used to proof that a language is not CFL language.

steps :

(1) assume L is a CFL. Let n be the natural no. obtained by using pumping leema

(II) choose $z \in L$ so that $|z| \geq n$. write $z = uvwxy$ using the pumping leema.

(III) find a suitable i so that $uv^iwx^iy \notin L$. This is a contradiction. so L is not CFL

Q. Show that $L = \{a^n b^n c^n : n \geq 0\}$ is not CFL.

Let we select a natural no. m and a string

$w = a^m b^m c^m$ here $|w| \geq m$

Now we can write $w = uvwxy$ such that $uv^i wx^i y \in L \;\; \forall i \geq 0$

where $|vwx| \leq m$

$|vx| \geq 1$

Case I:

Let $v, x \in$ only a's

Let $v = a, \; x = a$

then $\underbrace{aa(a)^i aa(a)^i \ldots a}_{m \text{ for } i=1} \; b^m c^m$

for $i = 2$ $\quad a^{m+2} b^m c^m \notin L$

Case II: Let $v, x \in$ only b's

$$\underbrace{\overbrace{aa\ldots aa}^{m}}_{u}\;\overbrace{\underbrace{bbb}_{v}\ldots\underbrace{.}_{w}\underbrace{bb}_{x}}^{m}\;\underbrace{\overbrace{cc\ldots c}^{m}}_{y}$$

now $uv^i wx^i y$ must $\in L \;\; \forall i \geq 0$

for $i = 2$

$uv^2 wx^2 y = a^m b^{m+2} c^m \notin L$

Case III Let $v, x \in$ only b's

$$\underbrace{a^m b^m}_{u}\;\overbrace{\underbrace{c}_{v}\underbrace{c}_{w}\underbrace{c}_{x}\underbrace{cc\ldots c}_{y}}^{m}$$

then $uv^i wx^i y$ (for $i=2$) $= a^m b^m c^{m+2} \notin L$

case (4) Let $v \in a's$ and $x \in b's$

$$\overbrace{\underbrace{aaa\ldots a}_{u}\underbrace{a}_{v}}^{m}\overbrace{\underbrace{bbb\ldots b}_{w}\underbrace{b}_{x}}^{m}\overbrace{\underbrace{ccc\ldots cc}_{y}}^{m}$$

then uv^iwx^iy (for $i=2$) $= a^{m+1}b^{m+1}c^m \notin L$

case (5) Let $v \in b's$ and $x \in c's$

$$\underbrace{\overbrace{aaa\ldots aa}^{m}\overbrace{bbb\ldots b}^{m}}_{u}\underbrace{b}_{v}\overbrace{\underbrace{ccc\ldots}_{w}\underbrace{c}_{x}\underbrace{c}_{y}}^{m}$$

then uv^iwx^iy (for $i=2$) $= a^mb^{m+1}c^{m+1} \notin L$

case (6) Let $v \in a's \;\&\; b's$, $x \in b's$

$$\overbrace{\underbrace{aaa\ldots a}_{u}a}^{m}\!\!\underbrace{\;}_{v}\overbrace{b\underbrace{bb\ldots b}_{w}\underbrace{b}_{x}}^{m}\overbrace{\underbrace{ccc\ldots cc}_{y}}^{m}$$

then uv^iwx^iy (for $i=0$) $= a^{m-1}b^{m-2}c^m \notin L$

case (7) Let $v \in a's$, $x \in a's \;\&\; b's$

case (8) Let $v \in b's \;\&\; c's$, $x \in c's$

case (9) Let $v \in b's$, $x \in b's \;\&\; c's$

in all these cases $uv^iwx^iy \notin L$ for some i

so the given language is not CFL.

Q using Pumping lemma, show that the language not CFL

(i) $L = \{a^n : n \text{ is a Prime}\}$

(ii) $L = \{0^i 1^j \mid j = i^2\}$

(iii) $L = \{0^m 1^n : m \neq n\}$

(iv) $L = \{a^n b^n c^i : n \leq i \leq 2n$

(v) $L = \{w w^R w : w \in (0,1)^*\}$