Grammar Definition: A (formal) grammar is a 4-tuple: G=(N,Σ,P,S) with the following meanings:

• N – set of nonterminal symbols and |N| < ∞

• Σ - set of terminal symbols (alphabet) and |Σ|

• P – finite set of productions (rules), with the propriety: P⊆(N∪Σ)∗ N(N∪Σ)∗ X(N∪Σ)∗

• S∈N – start symbol /axiom

Language derivated by the number

S -> ɛ

B->aB

Exercises:

1. G=(N,Σ,P,S); N={S,B,C}; ,Σ={a,b};

P: S->aB | bB

B-> aS

B->bC | b

S:S

Is aaabb ϵ L(G)

(1) (3) (1) (4) (6)

S=>aB=>aaS=>aaaB=>aaabC=>aaabb ϵ L(G)

Is aaba ϵ L(G)?

(1) (3) (2)

S=>aB=>aaS=>aabB !ϵ L(G)

2. G=(N,Σ,P,S); N={S,B,C}; ,Σ={a,b};

P: S->AB

A->aA | a

B->bB | b

S:S

L(G) ?

L(G) = {ab, aab, aabb, … }

L(G)= {a^m,b^n| m,n>=1}

3. L(G) = {a^mb^n | m >= 0, n > 0 }

G=?

L(G) = {b, ab, aabb, abb, …} – at least one b preceded by any number of a (including none)

G=(N,Σ,P,S); N={S,B}; Σ={a,b};

P: S-> a S | B

B-> bB | b

S: S

4. Construct a grammar that generates all arithmetic expressions containing 1 as operands and +,\*,(,,) as operators.

G=(N,Σ,P,S); N={S}; Σ={1,+,\*,(,)}

P: S -> (A) | A

A -> B+B | B\*B

B- > S | 1

S: S

OR

P: S - > 1 | S+S | S\*S | (S)

S: S

OR

P: E-> E+T | T

T -> T \* F | F

F -> (E) | 1

N = {E,T,F}; S: E

5. Given G=(N,Σ,P,S); N={S}; Σ={a,b,c}

P: S -> S | bc

Find L(G).

Let L ={bc|n}

Is L equal to L(G)?

(1) L include L(G)

- all sequence of that shape are generated by G

For any n in N bc is from L(G)

Proof by induction

P(0): included in L(G), S=>bc =

We suppose P(K) is true, then P(K+1) is true

P(k) true -> P(k): included in L(G)

\*

S=>

S=> => = (1) (induction hypo.)

* P(K+1) is true => (1) is validated

(2) L(G) include L

S => bc =

=>

=>

=>…

We notice that starting from S and using all grammar productions in all possible combinations, we only get sequence of terminals of shape where n is a natural number. It follows that the grammar does not generate anything else