Theory of Automata – Homework 5

Name – Akshay Kumar Singh

R11603620

1. Consider the grammar , where

Text, letter

Description automatically generated

a). Give a string of that can be produced by applying the rules at most 4 times

b). Same string can be derived in different ways, e.g., , . Give at least 2 distinct derivations for the string

c). For any , describe a derivation in of the string

Sol: (**a)** Length no more than 4:

S => AA=> aA=>aa

S=> AA=>aA=> abA=> aba

S=>AA=> aA =>aAb =>aab

S=> AA =>bAA =>baA=> baa

S =>AA =>bAA =>bAa=> baa

S => AA =>AbA => abA =>aba

S =>AA => AbA => Aba =>aba

S =>AA =>Aa=> aa

S =>AA =>Aa =>bAa => baa

S =>AA=> Aa => Aba => aba

S => AA =>AbA => abA => aba

S => AA => AbA => Aba =>aba

S => AA => AAb => aAb =>aab

S =>AA =>AAb=> Aab => aab

Applying the rules in different orders. strings that can be generated are: aa, aab, aba, baa.

**(b)** Note that A =>bA =>bAb => bab, and also that A =>Ab => bAb => bab. ( 8 distinct derivations):

S =>AA => AbA => AbAb => Abab =>\* babbab

S => AA => AAb => AbAb => Abab =>\* babbab

S => AA => bAA => bAbA =>babA =>\* babbab

S =>AA => AbA => bAbA => babA =>\* babbab

Where each of these four has 2 ways to reach babbab in the last steps.

**(c)** Producing a sequence in terms of m, n, p that will produce the string bmabnabp .

S => /\* by rule S → AA \*/

AA =>\* /\* by m applications of rule A → bA \*/

bmAA => /\* by rule A → a \*/

bmaA =>\* /\* by n applications of rule A → bA \*/

bmabnA =>\* by p applications of rule A → Ab \*/

bmabnAbp => /\* by rule A → a \*/

bmabnabp

Clearly this produces bmabnabp for each m, n, p.

1. Consider the grammar , where

Text, letter

Description automatically generated

Give a derivation of the string in .

Sol: Derivation of baabbb

S => bAb

=>bSSb

=>baAaSb

=>baSSaSb

=>baSaSb

=>baaSb

=>baabAbb

=>baabSSbb

=>baabSbb

=>baabbb

1. Show that the following languages are context-free by exhibiting context-free grammars generating each.

a).

b).

Sol: (**a)** : S -> aAB / AB

A -> aA / a

B -> bB / b / e

(**b)** : S -> ABC / e

A -> aA / a / e

B -> bB / b / e

C -> cC / c / e