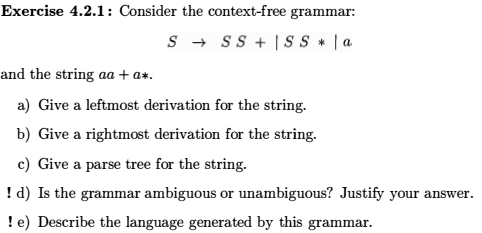
Max Bender

Compilers

Lab 3

Dragon:

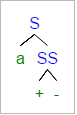


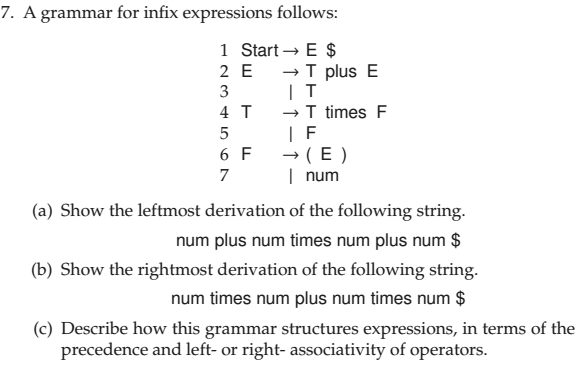
a)

* S S \*
* S S + S \*
* a a + S \*
* a a + a \*

b)

* S S \*
* S a \*
* S S + a \*
* S a + a \*
* a a + a \*

c) 



a)

* E $
* T plus E $
* F plus E $
* num plus E $
* num plus T plus E $
* num plus T times F plus E $
* num plus F times F plus E $
* num plus num times F plus E $
* num plus num times num plus E $
* num plus num times num plus T $
* num plus num times num plus F $
* num plus num times num plus num $

b)

* E $
* T plus E $
* T plus T $
* T plus T times F $
* T plus T times num $
* T plus F times num $
* T plus num times num $
* T times F plus num times num $
* T times num plus num times num $
* F times num plus num times num $
* Num times num plus num times num $

c) This grammer follows a pattern of gradually building out the statement by going from E -> T -> F. F eventually contains the num value which we see separating each of the plus and times statements. Because E and F also produce themselves we can build out the grammer as we go.

