Lab Three

shuhan Dong

shuhan.dong1@Marist.edu

March 13, 2021

1 Crafting a Compiler

exercises 4.7 (derivations)

a)

= Start

= 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)

=Start

=E $

=T plus E $

=T plus T $

=T plus T times F $

=T plus T 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)

Make operator precedence and left associative (operator times)or right associative(operator plus) to describe how this grammar constructs expressions. And precedence Order of operators : (), times, plus.

EXERCISES5.2C (RECURSIVE DESCENT PARSER PSEUDO CODE ONLY)

```
parseStart(){
parseValue()
match($)
}
parseValue(){
if(Token() == num){
match(num)
}
else{
match()
parseExpr()
match()
}
}
parseExpr(){
if(Token() == plus){
match(plus)
parseValue()
parseValue()
}
else{
match(prod)
parseValues()
}
```

}

parseValues(){

if(Token() == Value){

parseValue()

parseValues()

}

else

// epsilon production

}

2 Dragon

exercises 4.2.1 a, b, and c (derivations and a parse tree)

a)

S =>

SS*=>

SS+S*=>

aS+S*=>

aa+S*=>

aa+a*=>

b)

S =>

SS*=>

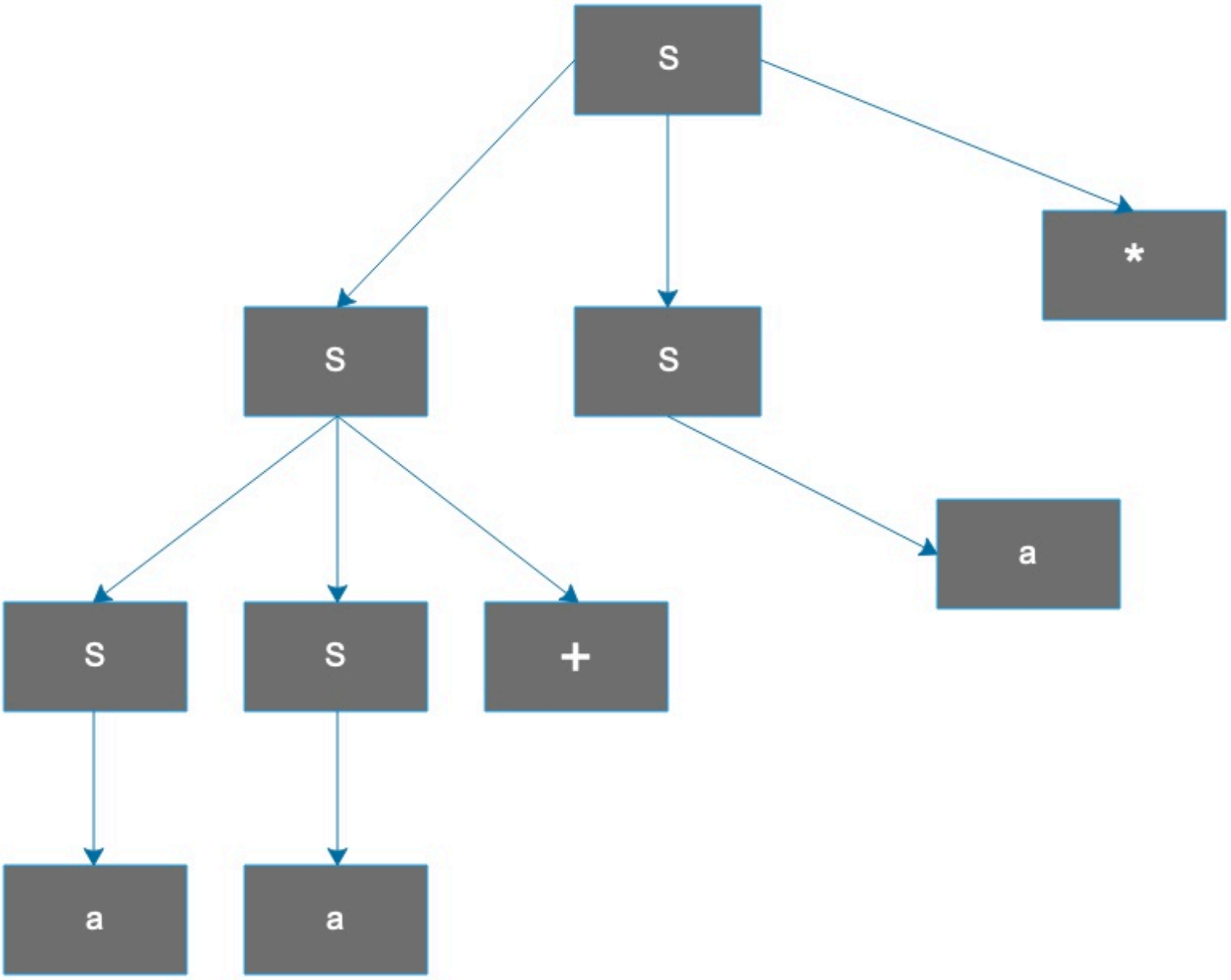
Sa*=>

SS+a*=>

Sa+a*=>

aa+a*=>

c)


S
S
S
*
S
S
+
a
a
a