CSCI 3155 Problem Set 2

Anthony Tracy

02/05/2018

1. ***Feedback: Complete survey linked from the moodle after completing this assignment. Any non-empty answer will receive full credit.***

   This has been done.

2. ***Grammars: Synthetic Examples***

   (a) Describe the Language defined by the following grammar:

   $S ::= ABA$
   $A ::= a|aA$
   $B ::= \epsilon|bBc|BB$

   This language will be a sequence of a's b's and c's and spaces in such a way that it will always start and end with the same number of a's. While in the center of the series will be made of b's on the left and c's on the right with spaces in there as chosen.

   (b) Now consider the next grammar:

   $S ::= AaBb$
   $A ::= Ab|b$
   $B ::= aB|a$

   Which of the Following sentences are generated for the above grammar?

   i. baab - Exists
   ii. bbbab - Cannot Exist

iii. bbaaaaa - Cannot Exist

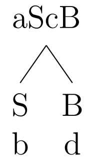
iv. bbaab - Exists

(c) Consider the following grammar:

$S ::= aScB|A|b$
$A ::= cA|c$
$B ::= d|A$

Which of the following sentenced are in the language generated by this grammar? Demonstrate the **parse tree** for each that are in the language.

i. abcd - Exists

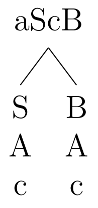


ii. acccbd - cannot exist

iii. acccbcc - cannot exist

iv. acd - cannot exist

v. accc - Exists



(d) Consider the following grammar:

$A ::= a|b|A \oplus A$

Show that this grammar is ambiguous.

Answer:

So this is ambiguous because it can become a or b just from the a—b, but then the $A \oplus A$ creates an issue, because it could be either of the A's which each can then restart the loop with plenty of other options, which are not concrete, they can be either or which creates ambiguity. There is more than one way to any one solution. Bad.

(e) Let us ascribe a semantics to the syntactic objects A specified in the above grammar from part d. In particular, let us write:

$A \Downarrow n$

for the judgment form that should mean A has a total n a symbols where n is the meta-variable for natural numbers. Define this judgment form via a set of inference rules. You may rely upon arithmetic operators over natural numbers. Hint: There should be one inference rule for each production of the non-terminal A (called a syntax-directed judgment form).

Answer:

Defining the judgment form as the following:

$\frac{(x \Downarrow n)(y \Downarrow n)}{x \oplus y \Downarrow n+m}$

3. ***Grammars: Understanding a Language.***

(a) Consider the following two grammars for expressions e. In both grammars, operator and operand are the same; you do not need to know their productions for this question.

e ::= operand | e operator operand
and
e ::= operand esuffix
esuffix ::= operator operand esuffix $|\epsilon$

i. Intuitively describe the expressions generated by the two grammars.

Answer:

Letting operator as something such as a derivative d/dx, and the operand being what the operator acts on, i.e - f(x): Then the first grammar would either return just the function, or value being acted on, or expand out to be the function, or value, multiplied against the operator acting on the operand to the power of the number of times the loop runs.

The second is very similar, it would again always have an operand followed by nothing or the operator acting on the operand to the power of n loops. This just terminates using whitespace.

ii. Do these grammars generate the same or different expressions? Explain.

Answer:

These would generate the same languages, they just differ based on how they terminate. The first builds a list of operator acting on operand, until terminating by tacking an operand to the left most of the string. While the second will always start with the leftmost operand, then keep adding an operator acting on an operand to the right most of the series, until terminating with whitespace at the right most end.

(b) Write a Scala expression to determine if – has higher precedence than $<<$ or vice versa. Make sure that you are checking for precedence in your expression and not for left or right associativity. Use parentheses to indicate the possible abstract syntax trees, and then show the evaluation of the possible expressions. Finally, explain how you arrived at the relative precedence of – and $<<$ based on the output that you saw in the Scala interpreter.

Answer:

In the end – has a higher precedence than $'<<'$ which it may also be good to note that scala bases precedence on the first character of the operator.

(c) Give a BNF grammar for floating point numbers that are made up of a fraction (e.g.,5.6 or 3.123 or -2.5) followed by an optional exponent (e.g., E10 or E-10). The exponent, if it exists, is the letter E followed by an integer. For example, the following are floating point numbers: 3.5E3, 3.123E30, -2.5E2, -2.5E-2, and 3.5. The following are not examples of floating point numbers: 3.E3, E3, and 3.0E4.5. More precisely, our floating point numbers must have a decimal point, do not have leading zeros, can have any number of trailing zeros, non-zero exponents (if it exists), must have non-zero fraction to have an exponent, and cannot have a – in front of a zero number. The exponent cannot have leading zeros. For this exercise, let us assume that the tokens are characters in the following alphabet Σ:

$$\Sigma = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, E, -, .$$

Your grammar should be completely defined (i.e., it should not count on a non-terminal that it does not itself define.)

Answer:

$F ::= n.nEn | -n.nEn | n.nE-n | -n.nE-n$
$n ::= n0 | nInt | Intn$
$Int ::= 1|2|3|4|5|6|7|8|9$

4. ***JavasScripty Interpreter: Booleans, Strings, Varialbe Binding, and Conversions.***

This code has been written and has been submitted.