Quinn Roemer

CISP - 440

Assignment 15

12/13/2018

Part 0 - Languages and Grammars.

Description:

The goal for this assignment is to demonstrate my knowledge on languages and grammars. To do this, I was to perform a number of practice problems defined by the professor. Note, each question will be listed below the problem.

Problem 2:

Determine whether the given grammar is context-sensitive, context-free, regular, or none of these. Give all characterizations that apply.

T = {a, b, c}
N = {ơ, A, B}

Productions:

ơ -> AB	AB -> BA	A -> ơA
B -> Bb	A -> a	B -> b

Start = ơ

This grammar defines a **context free** language. This is because it fulfills the three requirements for such a language. It has choice, a “B” can either become a “b” or a “Bb.” It has Recursion, “B” goes to “Bb.” And, it has a base case. Eventually the grammar would terminate because no non-terminals would be left.

Problem 3:

Determine whether the given grammar is context-sensitive, context-free, regular, or none of these. Give all characterizations that apply.

T = {a, b}
N = {ơ, A, B}

Productions:

ơ -> A	ơ -> AAB	Aa -> ABa	
A -> aa	Bb -> ABb	AB -> ABB	B -> b

Start = ơ

This grammar is **context sensitive**. This is because some substitutions rely on what is around to apply. For example “Aa” to “ABa” requires an “Aa” to be there (context) before the “B” can be wedged in between.

Problem 8:

Show that the given string “a” is in L(G) for the given grammar G by giving the derivation of “a.”

G = problem 2
“a” = abab

<u>σ</u> -> <u>A</u>B -> <u>σ</u>AB -> <u>A</u>BAB - > a<u>B</u>AB -> ab<u>A</u>B -> aba<u>B</u> -> abab

Note: The underlined portion is where the sub is being performed.

Problem 11:

Show that the given string “a” is in L(G) for the given grammar G by giving the derivation of “a.”

T = {a, b}
N = {S, A, B}

Productions:

S -> bS	S -> aA	S -> a
A -> aS	A -> bB	
B -> bA	B -> aS	B -> b

Start = S

“a” = abaabbabba

<u>S</u> -> a<u>A</u> -> ab<u>B</u> -> aba<u>S</u> -> abaa<u>A</u> -> abaab<u>B</u> -> abaabb<u>A</u> -> abaabba<u>S</u> -> abaabbab<u>S</u> -> abaabbabb<u>S</u> -> abaabbabba

Note: The underlined portion is where the sub is being performed.

Problem 12:

Write the grammars of Examples 10.3.4 and 10.3.9 and Exercises 1-4 and 6 in BNF.

10.3.4

<Ơ> ::= ba | a <S>

<S> ::= b <S> | b

10.3.9

<Ơ> ::= a <A> <B> | a <B>

<A> ::= a <A> <C> | a <C>

<B> ::= <D> c

<D> ::= b

<C> <D> ::= <C> <E>

<C> <E> ::= <D> <E>

<D> <E> ::= <D> <C>

<C> c ::= <D> cc

Exercise 1

<Ơ> ::= b <Ơ> | a <A> | b

<A> ::= a <Ơ> | b <A> | a

Exercise 2

<Ơ> ::= <A> <B>

<A> ::= a <A> | a

<B> ::= <B> b | b

<A> <B> ::= <B> <A>

Exercise 3

<Ơ> ::= <A> | <A> <A> <A> <B>

<A> ::= aa

<B> ::= b

<A> a ::= <A> <B> a

<B> b ::= <A> <B> b

<A> <B> ::= <A> <B> <B>

Exercise 4

<Ơ> ::= <B> <A> <B> | <A> <B> <A>

<A> ::= <A> <B> | a <A> | a <B>

<B> ::= <B> <A> | b

Exercise 6
<Ơ> ::= <A> <A> <Ơ>
<A> <A> ::= <B>
<B> ::= b <B>
<A> ::= a

Problem 16:

Write a grammar that generates the strings having the given property.

Property: Strings over {a, b} ending with “ba.”

T = {a, b}
N = {S, A, B}

Productions:

S -> ABS	S -> BAS	S -> ba
A -> aA	A -> aB	A -> a
B -> bB	B -> bA	B -> b

Starting symbol = S

The above grammar is capable of only making strings that end with “ba.”

Problem 17:

Write a grammar that generates the strings having the given property.

Property: Strings over {a, b} containing “ba.”

T = {a, b} and N = {S, A, B}

Productions:

S -> AsB	S -> BsA	S -> ba
A -> aA	A -> aB	A -> a
B -> bB	B -> bA	B -> b

Starting symbol = S

The above grammar is capable of only making strings that contain “ba.”

Problem 19:

Write a grammar that generates the strings having the given property.

Property: Integers with no leading 0's.

T = {-integers, +integers, 0}
N = {S, A, B}

Productions:

S -> SAB	S -> SBA	S -> -integers	S -> +integers
A -> +integers	A -> 0		
B -> +integers	B -> 0		

Starting symbol = S

The above grammar is capable of making positive or negative integers that do NOT start with 0.

Problem 21:

Write a grammar that generates the strings having the given property.

Property: Exponential numbers (numbers including floating point)

T = {-integers, +integers, 0, point, exponent}
N = {S, A, B}

Starting symbol = S

Productions:

S -> A point B	S -> A exponent A	S-> A point B exponent A
A -> -integers	A -> +integers	A -> 0
B -> +integers	B -> 0	

This above grammar is capable of producing a float, or exponential number that is either negative or positive and is raised to some negative or positive power. Note, it is not possible to create a number such as 9.-12 since a negative sign is not allowed past the point.

Problem 28:

Each grammar is proposed as generating the set L of strings over {a, b} that contain equal numbers of a’s and b’s. If the grammar generates L, prove that it does so. If the grammar does not generate L, give a counterexample and prove that your counterexample is correct. In each grammar the starting symbol is S.

S -> abS | baS | aSb | bSa | ƛ

This grammar is only capable of producing a set of strings that contain an equal number of a’s and b’s. This is because for every substitution of S one “a” and one “b” are introduced. Never is a letter added by itself. Thus, the string is always balanced.

Conclusion

This assignment was a little bit confusing. Primarily because I was slightly confused on how to write such grammars. It would’ve been helpful in class to do more examples closer to that which we see in the book. Otherwise, I found this homework to be interesting. The idea of grammars defined by certain logical rules intrigues me. Looking forward to our next and last assignment!