Extended BNF Grammar Assignment

Assignment objective:

To provide practice in rewriting BNF grammar rules into Extended BNF grammar rules as well as practice in removing Direct Left Recursion.

Assignment:

Download the assignment's Microsoft Word document answer file provided along with this assignment in Blackboard, having the name **YourLastName_YourFirstName_Assignment2.docx**.

Every student at Queens College has access to Microsoft Office 365 via their QC e-mail account: https://qmail.cuny.edu. You do not even have to install Microsoft Word on your computer to use it.

Rename the Microsoft Word document by replacing the text **YourLastName** in the filename with your actual last name and replacing the text **YourFirstName** in the filename with your actual first name. Make sure you type your full name and your CUNYFirst ID number on the top line of page 1 in the space provided.

Rewrite the ADALS1 BNF Grammar into Extended BNF. Not every rule needs to be rewritten. Some of the rules have Direct Left Recursion. Therefore, you need to rewrite the rules to eliminate the Direct Left Recursion and then you can rewrite those rules in Extended BNF.

After you complete the assignment, follow the submission instructions on page 3 of this assignment.

Grading Criteria:

This assignment is worth 20 points.

1. Proper submitted file format: If the file submitted is not a pdf file, then 1 point is subtracted from the grade.
2. Proper file name: If the file submitted does not have the proper file name format as described on the Page 1 in this assignment description, then 1 point is subtracted from the grade.
3. Your full name and CUNYFirst ID number on page 1: If your full name and CUNYFirst ID number are not at the top line of page 1 in your answer document, then 1 point is subtracted from the grade.
4. Late submission penalty: Assignments submitted after the due date are subjected to a 2-point deduction for each day late.

Submission Instructions:

IMPORTANT!:
Make sure you have properly renamed your answer document file as described on Page 1 in this assignment description.
After you finish entering your answer and save your document, save your document as a pdf file. It is the pdf file that you will submit.

Once you submit your assignment you will not be able to resubmit it!
Make absolutely sure the assignment you want to submit is the assignment you want graded.
There will be **NO** exceptions to this rule!

You will submit your pdf file via your CUNY Blackboard account.
This is the only accepted submission method.

Follow these instructions:

Log onto your CUNY Blackboard account.
Click on the CSCI 316 course link in the list of courses you are taking this semester.
Click on **Assignments** in the red area on the left side of the webpage.
You will see **Extended BNF Grammar Assignment**.
Click on the assignment.
Upload your pdf file and then click the submit button to submit your assignment.

Due Date: Submit this assignment on or before 11:59 p.m. Monday, November 29, 2021.

Late submission penalty: assignments submitted after the due date are subjected to a 2-point deduction for each day late.

Late submission policy: you **CAN** submit your assignment early, before the due date. You are given plenty of time to complete the assignment well before the due date. Therefore, I do **NOT** accept any reason for not counting late points if you decide to wait until the due date (and the last possible moment) to submit your assignment and something happens to cause you to submit your assignment late. I only use the date submitted, ignoring the time as well as Blackboard's late submission label.

Name: Calvin Thang CUNYFirst ID Number: 23733402

Type your name and CUNYFirst ID # above.

Ada Language Subset 1 (ADALS1) Grammar

```
<S> → procedure IDENT is <DECPART> begin <SEQOFSTMT> end ; EOI

<DECPART> → ε
<DECPART> → <OBJECTDEC> <DECPART>

<OBJECTDEC> → <IDENTLIST> : boolean ;
<OBJECTDEC> → <IDENTLIST> : integer ;

<IDENTLIST> → IDENT
<IDENTLIST> → IDENT , <IDENTLIST>

<SEQOFSTMT> → <STATEMENT>
<SEQOFSTMT> → <STATEMENT> <SEQOFSTMT>

<STATEMENT> → null ;
<STATEMENT> → IDENT := <EXPRESSION> ;
<STATEMENT> → if <CONDITION> then <SEQOFSTMT> end if ;
<STATEMENT> → if <CONDITION> then <SEQOFSTMT>
                              else <SEQOFSTMT> end if ;
<STATEMENT> → while <CONDITION> loop <SEQOFSTMT> end loop ;
<STATEMENT> → get ( <IDENTLIST> ) ;
<STATEMENT> → put ( <IDENTLIST> ) ;
<STATEMENT> → newline ;

<CONDITION> → <EXPRESSION>

<EXPRESSION> → <SIMPEXPR>
<EXPRESSION> → <SIMPEXPR> = <SIMPEXPR>
<EXPRESSION> → <SIMPEXPR> /= <SIMPEXPR>
<EXPRESSION> → <SIMPEXPR> < <SIMPEXPR>
<EXPRESSION> → <SIMPEXPR> <= <SIMPEXPR>
<EXPRESSION> → <SIMPEXPR> > <SIMPEXPR>
<EXPRESSION> → <SIMPEXPR> >= <SIMPEXPR>

<SIMPEXPR> → <SIMPEXPR> + <TERM>
<SIMPEXPR> → <SIMPEXPR> - <TERM>
<SIMPEXPR> → <TERM>

<TERM> → <TERM> * <PRIMARY>
<TERM> → <TERM> / <PRIMARY>
<TERM> → <TERM> rem <PRIMARY>
<TERM> → <PRIMARY>
```

Recall: EBNF extensions

{ } – repeat indefinitely or leave out altogether

[ ] – optional

(|) – must choose one out of multiple choices

<PRIMARY> → (<EXPRESSION>)
<PRIMARY> → IDENT
<PRIMARY> → NUMLIT
<PRIMARY> → true
<PRIMARY> → false

<S> → procedure IDENT is <DECPART> begin <SEQOFSTMT> END; EOI

<DECPART> → <OBJECTDEC> <DECPART>
| ε

<OBJECTDEC> → <IDENTLIST> : (boolean | integer) ;

<IDENTLIST> → IDENT {, <IDENTLIST>}

<SEQOFSTMT> → <Statement> {<seqofstmt>}

<Statement> → null ;
" " → ident := <expression>
" " → if <cond> then <seqofstmt> [else <seqofstmt>] end if;
" " → while <cond> loop <seqofstmt> end loop ;
" " → (get | put) (<IDENTLIST>) ;
" " → newline ;

<Condition> → <expression>

<expression> → <simpexpr> [(= | /= | < | <= | > | >=) <simpexpr>]

<simpexpr> → <term> {(+|-) <term>}

<term> → <primary> {(* | / | rem) <primary>}

<primary> → (<expression>)
" " → IDENT
" " → NUMLIT
" " → true
" " → false