CS4450 Homework #2: Context-Free Grammars (CFGs)

In this assignment, you will define context-free grammars (CFGs) for four different language descriptions (with a fifth bonus question) using ANTLR4. You will modify the provided .g4 files to create grammars that match the requirements for each question. You will then test your grammars using the provided Python script (HW_02.py) to ensure they produce the correct results.

Objective

- Implement CFGs for four language descriptions and one bonus question.
- Modify the provided .g4 files to match the requirements of each question.
- Use ANTLR4 to generate parsers test your grammars with the provided Python script.

Instructions

1. Install ANTLR4

- Please refer to the material posted on Canvas by Dr. Ekin for detailed instructions on installing ANTLR4.
- If you encounter issues with the ANTLR4 installation, please contact the TAs for assistance.

2. Editing the .g4 Files

1. Open the .g4 Files (Q1.g4 to Q5.g4) are provided to you with a basic structure.
2. For each question, modify the start rule and add production rules to define the CFGs for each language description given in the assignment PDF.

- Example Structure:

```
grammar Q1; // This should match the file name Q1.g4

start: // Define your grammar rules here

;
```

*Do Not Remove Existing Content

3. **Using ANTLR4 to Generate Parser and Lexer**:

- Open a terminal or command prompt. Navigate to the directory containing your .g4 file.

- Run the following command using ANTLR's Python runtime to generate the parser and lexer files:

```
antlr4 -Dlanguage=Python3 Q1.g4
```

(Replace Q1 with Q2, Q3, etc., as needed)

This command will generate Python files for the lexer and parser (e.g., Q1Lexer.py and Q1Parser.py).

4. Testing Your Grammars with HW_02.py

1. Place All Files in the Same Directory: (.g4 files, generated Python files, and HW_02.py script)

2. Run the Python Script: `python HW_02.py`

3. Check the Output:

- The script will test your grammar against predefined valid and invalid cases.

- Review the output to see which test cases passed and which failed.

- Use this feedback to adjust and refine your grammar if necessary.

4. Edge Case Testing: Ensure you test your parser for all possible edge cases. **Note that additional test cases may be used during grading.**

5. Submitting Your Assignment

1. After verifying that your grammar works correctly, submit the modified .g4 file for each question.

2- Do not modify the provided test cases in `HW_02.py`. Your grammars should match the test cases as described in the assignment.

Notes

- Make sure that your grammar is unambiguous and covers all specified requirements.

- Ensure that your grammar does not accept invalid strings.

- Remember that during grading, additional test cases may be used to evaluate your grammars.

Grading Criteria

- Correctness of the grammar based on the requirements of each question.

- Passing all valid and invalid test cases as provided in `HW_02.py`.

- Consideration of edge cases as part of your testing.

All the best!