COMP 2049 Languages and Computation Coursework: Statements in programming languages

Overview

In most of the modern programming languages, a statement is defined as a syntactic unit of the language that expresses some actions to be carried out. Like sentences in natural languages, statements are the fundamental building blocks of programming languages. One of the compound statements we have learned in this module is the if-statement.

Given the following syntactic grammar of the if-statement:

IfStatement :

if (Expression) Statement

if (Expression) Statement *else* Statement

We could design a *contex-free grammar* $G_1 = (V_1, T_1, S_1, P_1)$ that generates the language of if-statements:

- V_1 = { S_1, E, O }

- T_1 = { *if*, *else*, *exp*, *stmt*, (,)}

- P_1 is defined as:

$$
\begin{aligned}
S_1 &\rightarrow \textit{if}\ \ (E)\ \ S \\
S_1 &\rightarrow \textit{if}\ \ (E)\ \ \textit{else}\ \ S \\
S &\rightarrow S_1 \mid O \\
E &\rightarrow \textit{exp} \\
O &\rightarrow \textit{stmt}
\end{aligned}
$$

Where S_1 is the start symbol representing the *if statement*, E and S denote an expression and a statement respectively, and O is a statement that is not an if statement. For simplicity, in all the following tasks, we assume that *exp* is the only expression we know, and *stmt* represents a valid 'other statement'.[1]

In this coursework, you are required to extend the above definition to include additional types of statements in programming languages. You are expected to extend the grammar and implement it using JFLAP. For all the tasks below, ensure that you are using JFLAP version 7.1, which can be downloaded from the LAC Moodle page.

[1] In this CW, we focus on designing grammars for particular types of statements in programming languages. Thus, we will not define grammars for expressions and all other types of statements.

While Statement

Task 1. Given the following syntactic grammar for the while-statement, design a *context-free grammar* $G_2 = (V_2, T_2, S_2, P_2)$ that generates the language of while-statements:

WHILESTATEMENT :

while (EXPRESSION) STATEMENT

Assume that S_2 is the start symbol representing a while statement, E is the variable for an expression, S is the variable for a statement, and O is the variable for a statement that is not a while statement. Complete the *context-free grammar* below for while statements:

- V_2 = {...}

- T_2 = {...}

- P_2 is defined as:

$$S_2 \quad \rightarrow \quad ?$$
$$\ldots$$

Your solution must include:

1. The complete definition of V_2, T_2 and P_2.
2. A description of each new variable in V_2, if any exists.

For Statement

Task 2. Given the following syntactic grammar for the for-statement, design a *context-free grammar* $G_3 = (V_3, T_3, S_3, P_3)$ that generates the language of for-statements. Note that the subscript 'opt' indicates optional, i.e, A_{opt} signifies that the appearance of A is optional.

FORSTATEMENT :

for(FORINIT$_{opt}$; EXPRESSION$_{opt}$; FORUPDATE$_{opt}$) STATEMENT

Assume that S_3 is the start symbol representing a for statement, E is the variable for an expression, S is the variable for a statement, O is the variable for a statement that is not a while statement. For simplicity, 'init' is used to represent the for-initialization, and 'update' represents the for-update. Complete the *context-free grammar* below for for statements:

- V_3 = {...}

- T_3 = {...}

- P_3 is defined as:

$$S_3 \quad \rightarrow \quad ?$$

$$\ldots$$

Your solution must include:

1. The complete definition of V_3, T_3 and P_3.
2. A description of each new variable in V_3, if any exists.

Unambiguous Grammar and JFLAP

Task 3. In this task, you are required to design an unambiguous *context-free grammar* $G = (V, T, S, P)$ for representing statements in programming languages. In particular, the *if-statement*, *while-statement* and *for-statement* must all be included. Assume that E is the variable for an expression, S is the variable for a statement, O is the variable for a statement that is not a while statement, a for statement or an if statement. All other types of statements are represented by $stmt$ and all the expressions are denoted as exp. For for-initialization and for-update, $init$ and $update$ are used respectively as in Task 2.

- $V = \{...\}$

- $T = \{...\}$

- P is defined as:

$$S \quad \rightarrow \quad ?$$

$$\ldots$$

Your solution must include:

1. The complete definition of V, T and P.
2. A description of each new variable in V if any exists.

Task 4. *Implement the grammar G using JFLAP.* Your solution must accept all valid statements, reject all other inputs, and avoid any λ-productions or unit-productions (see Figure 1 as an example test case).

Figure 1: Some sample input and the corresponding results

Input	Result
if(exp)stmt	Accept
while(exp)stmt	Accept
if(exp)if(exp)	Reject
for()stmt	Reject
for(;;)stmt	Accept
for(init;exp;update)stmt	Accept
if(exp)if(exp)stmtelsestmt	Accept
if(exp)while(exp)if(exp)stmtelsestmt	Accept
while(if(exp)stmt)stmt	Reject

Submission

You must submit one zip file which contains one PDF file and one JFLAP file, named according to the following templates:

- The PDF file must be named:

  ID_Surname_Firstname.pdf

- The JFLAP file must be named:

  ID_Surname_Firstname.jff

- The zip file must be named:

  ID_Surname_Firstname.zip

  Where ID is your students ID, Surname and Firstname are your family name and first name respectively.

- Release date: Mondy, 7^{th} of April, 2025
- Deadline: Friday, 18^{th} of April, 2025, 17:00
- Weight: 15% of the module mark
- How to submit: via Moodle

Marking Scheme

Correctness: Correct answers for the four tasks contribute to 95% of the total mark, as follows:

- **Task 1:** 15%
- **Task 2:** 25%
- **Task 3:** 35%
- **Task 4:** 20%

Format:

1. The zip file, pdf file and the jff file must be named according to the templates given above. (5%).

Late Submissions: The standard University penalty for late submission is applied, i.e., 5% absolute standard University scale per day, until the mark reaches zero.

Use of AI Technologies: If you use AI tools other than those specified in the assignment brief, e.g., chatGPT, 100% deduction