German University in Cairo
Department of Computer Science
Assoc. Prof. Haythem O. Ismail

CSEN1002 Compilers Lab, Spring Term 2023
Task 9: ANTLR Parsing I

Due: Week starting 20.05.2023

1 Objective

For this task you will use ANTLR (`www.antlr.org`) to implement an SDD for the following problem. ANTLR documentation is available here:

`https://github.com/antlr/antlr4/blob/master/doc/index.md`

2 Requirements

- You are required to use ANTLR to implement the SDD appearing below for a CFG that generates the language $\mathtt{a}^*\mathtt{c}^*\mathtt{b}^*$.

$$
\begin{array}{lll}
S & \longrightarrow & ACB \qquad C.l = A.2n;\ C.u = A.3n \\
 & & \qquad\qquad C.ilf = 0;\ C.iuf = 1 \\
 & & \qquad\qquad S.check = C.slf * C.suf * equals(A.n, B.n) \\
A & \longrightarrow & \mathtt{a}A_1 \qquad A.n = A_1.n + 1;\ A.2n = A_1.2n * 2;\ A.3n = A_1.3n * 3 \\
A & \longrightarrow & \varepsilon \qquad A.n = 0;\ A.2n = 1;\ A.3n = 1 \\
B & \longrightarrow & \mathtt{b}B_1 \qquad B.n = B_1.n + 1 \\
B & \longrightarrow & \varepsilon \qquad B.n = 0 \\
C & \longrightarrow & \mathtt{c}C_1 \qquad C_1.l = C.l;\ C_1.u = C.u \\
 & & \qquad\qquad C_1.ilf = C.ilf;\ C_1.iuf = C.iuf \\
 & & \qquad\qquad C.m = C_1.m + 1 \\
 & & \qquad\qquad C.slf = C_1.slf + equals(C.l, C.m) \\
 & & \qquad\qquad C.suf = C_1.suf - equals(C.u, C_1.m) \\
C & \longrightarrow & \varepsilon \qquad C.m = 0;\ C.slf = C.ilf;\ C.suf = C.iuf
\end{array}
$$

- The start variable S has an attribute *check* whose value is 1 if the generated string is of the form $\mathtt{a}^n\mathtt{c}^m\mathtt{b}^n$ with $2^n \leq m \leq 3^n$, for some $n \geq 0$, and is 0 otherwise.

- **The only operations allowed on attributes are assignments, additions, multiplications, and equality checks; an equality check is an expression of the form $equals(x, y)$ whose value is 1 if x is equal to y and is 0 otherwise.**

- The provided method `sCheckValue` uses the ANTLR grammar to get the value of *S.check* for a given input string. For example, for the string `aaccccccbb`, `sCheckValue` returns 1; and returns 0 for the string `aacbb`.

- Important Details

- Your implementation should be done within the template file uploaded to the CMS.
- You are not allowed to change the grammar name, the rule name “`s`” or attribute “`check`”.
- You are allowed to write as many helper parser and lexer rules within the same grammar file (if needed).
- Public test cases have been provided on the CMS for you to test your implementation.
- Please ensure that the public test cases run correctly without modification before coming to the lab to maintain a smooth evaluation process.
- A Java file is provided in order to easily test your grammar with custom strings in addition to the public test cases.
- Private test cases will be uploaded before your session and will have the same structure as the public test cases.

3 Evaluation

- Your SDD will be tested using ten inputs.
- You get one point for each correct output; hence, a maximum of ten points.

4 Online Submission

- You should submit your code at the following link.

  `https://forms.gle/q24XmZ7BZV4a8bgH6`

- Submit one file “`Task9.g4`” containing the grammar.
- **Online submission is due by the end of your lab session.**