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

Compilers, Spring Term 2025
Assignment 2
Due: May 20 at 23:59

1 Objective

For this task you will use ANTLR to implement an SDT or an SDD to count the number of extrema in a non-empty sequence of digits. Digit d_i is an extremum in a sequence $d_1, \ldots, d_n$ of digits if either $d_{i-1} < d_i > d_{i+1}$ or $d_{i-1} > d_i < d_{i+1}$. (Note that d_1 and d_n cannot be extrema.)

2 Requirements

- The grammar of your SDT/SDD should generate all strings representing non-empty sequences of digits.
- A non-empty sequence of digits is represented by a string of the form "$d_1, d_2, \ldots, d_n$", where d_i is a (decimal) digit.
- For example, the following are representations of non-empty sequences of digits.
  (a) `1,4,2,5,3`
  (b) `9,8,9,7,6`
  (c) `2,4,5`
  (d) `1`
  (e) `5,0,0,7,3`
- In your SDT/SDD, the start variable `s` should have an attribute `val` whose value is the number of extrema in the input sequence.
- In the example sequences above, `val` should be 3 for sequence (a), 2 for sequence (b), 0 for sequences and (c) and (d), and 1 for sequence (e).
- **The only operations allowed on attributes are assignments, equality checks (`==`, `!=`), relational checks (`<`,`>`, `<=`, `>=`), and arithmetic operations (`+`, `-`, `*`).**
- 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 "`val`".
  - You are allowed to write as many additional parser and lexer rules within the same grammar file (if needed).
  - A Java file is provided in order to easily test your grammar with custom strings.

3 Online Submission

- You should submit your code at the following link.

  `https://forms.gle/Kn5QGK6iHAhbBjME9`

- Submit one file "`Assignment2.g4`" containing the grammar.