CS350 2022-23: Project

Due date: November 15, 2022

The project is worth 100 points. Each question is worth 25 points. It may be helpful to read, or revise the concept of "recursive descent parsers". You can find this, for example, in "Principles of Compiler Design" by Aho, Ullman or "Compilers: Principles, Techniques and Tools" by Aho, Ullman or Sethi, or online.

1. Implement a grammar checker for the following fully parantelesized form of the λ calculus.

The allowed variables are single lowercase English letters - a, b, c etc. You can assume that these are sufficient - very long expressions will not be given.

The grammar is as follows. Note that λ is represented by \setminus for easier parsing.

 $\begin{array}{l} \lambda - \mathrm{term} ::= \mathrm{variable} | \\ & (\backslash variable \cdot \lambda - \mathrm{term}) \\ & [\lambda - \mathrm{term}] [\lambda - \mathrm{term}] \end{array}$

Given an expression, output whether it is a valid λ -term or not.

2. Given a valid λ -term, output the list of free variables.

3. Implement the algorithm to substitute a λ -term replacing all free occurrences of a variable in another λ -term.

4. Implement the β -reduction rule.