

Computer Science 401
Automata and Formal Languages
Fall Quarter, 2015
Second Programming Assignment
Due: Wednesday, November 4

Implement a Scheme function called RE? that takes a list of symbols as input and returns true (#t) if the sequence of symbols is a legal regular expression and false (#f) otherwise.

Because parentheses and other symbols have special meaning in the Scheme programming language, we will work with a definition of regular expressions using a different set of symbols. Here is our new definition:

1. **lamb**, **empty** and any atom other than **o** is a regular expression (thus, **lamb** stands for Λ , **empty** stands for \emptyset , **o** stands for the concatenation operator, \cdot) and
2. If **R** and **S** are regular expressions then **/ R &**, **R + S**, **R o S** and **R *** are regular expressions. That is, **/** stands for a left parenthesis (**,** and **&** stands for a right parenthesis (**,**), and **o** stands for the concatenation operator.

In this revised notation, the regular expression

$$\Lambda + a \cdot b + a \cdot b \cdot a \cdot b \cdot (a \cdot b)^*$$

would be represented as the list

$$(\text{ lamb} + \text{ a o b} + \text{ a o b o a o b o} / \text{ a o b} / \& *)$$

and

$$\Lambda + a \cdot b + (a + b) \cdot a \cdot b \cdot (a \cdot b)^*$$

would be written as

$$(\text{ lamb} + \text{ a o b} + / \text{ a} + \text{ b} \& \text{ o a o b o} / \text{ a o b} \& *)$$

(Note that in this modified representation each constituent is always separated by a space.)

The input

$$(\text{ lamb} + \text{ a o b} + \text{ a o b o a o b o} / \text{ a o b} \& *)$$

should result in #t, confirming the syntactic correctness of the sequence and the input because it is equivalent to

$$\Lambda + a \cdot b + a \cdot b \cdot a \cdot b \cdot (a \cdot b)^*$$

and the input

$(\text{lamb} + a o b + a o b o a o b o / * \& *)$

should result in **#f** (because it would be equivalent to $\Lambda + ab + abab(*)^*$)

The purpose of the following assignment is motivational. Although seemingly straightforward, it can be quite difficult to write a program that responds correctly to all cases; however, later on in this class you will learn techniques that make the problem quite trivial to solve.

To implement this function, use the Racket language. A properly commented source program should be submitted to Canvas.