GR-512

A FCFS Approach for Order of Operations in Arithmetic Formalisms in Programming Languages

INTRO/ABSTRACT

Most conventional algebras, most make use of PEMDAS. Now suppose that a language can only perform binary operations on two logical variables L ∧ R. Given any arithmetic expression how many variables outside of two logical variables L ∧ R are needed to express the expression?

METHODS

To begin to answer the question of computational powers of a programing Language we need a way to evaluate it. To do this we present a new formalism:

- Definitions for Preemptive and Non-Preemptive Algebras
- New Model of Computation the LRT Machine
- New Classifications for expressions

RESULTS

Theorem 1. Given a preemptive language L, every expression e ∈ E(L) can be expressed as a sequence S of expressions with at most one operation such that the final expression in S is equivalent to e

Theorem 2. Given a preemptive language L, every expression e ∈ E(L) of class of n can be computed on a corresponding LRT machine with n auxiliary registers

Created a new way to understand the structural limitations of programing lnaguge.

Future Results

With this new understanding of the limitations of program languages I plan to extended the formalism to other functions of common programing language paradigms

What Problems are Solvable using a given Programing Language?

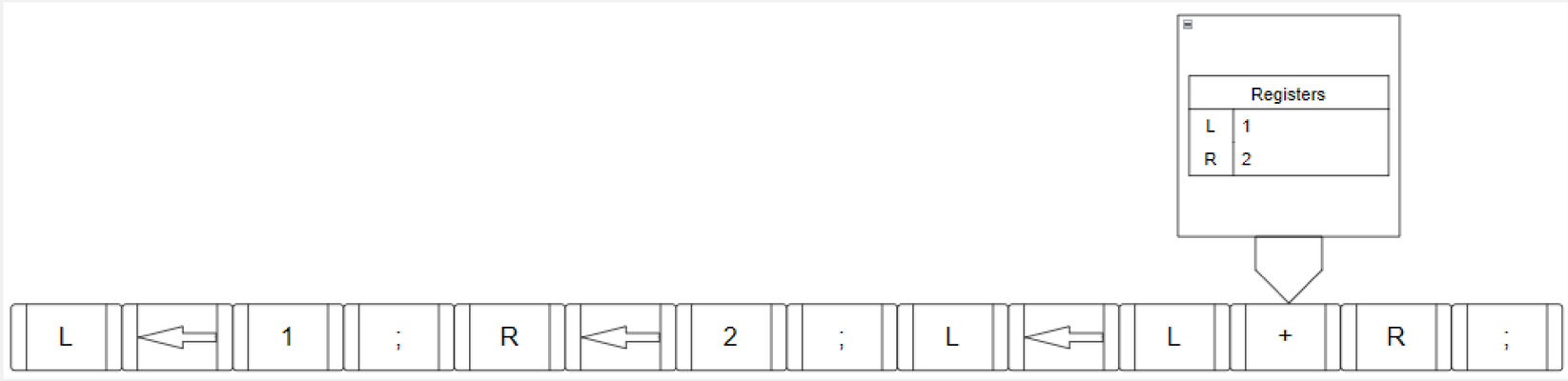

Fig.1 An abstract LRT machine computing the expression 1+2

My LinkedIn account

KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING

Author(s) Ethan Hunt
Advisors(s) Dr. Rifatul Islam