

The efficiency of primitive recursive functions: a programmer's view

Armando B. Matos¹

^a...

Keywords: primitive recursion, complexity of Loop programs, operational semantics

Email address: armandobcm@yahoo.com (Armando B. Matos)
¹...(www...)

1. Introduction

Extending the language Loop or the “classical” definition of p.r. functions

Purpose of the paper

Related work

Paper organization

2. Notation and preliminaries

Functions

Orders

Languages

3. The languages

3.1. Loop

3.1.1. The function computed by a program

3.1.2. Alternatives to the basic Loop language. Minimality

3.1.3. Execution time

3.2. Loop with additional instructions

3.2.1. Loop plus break, “Loop+break”

3.2.2. Loop plus conditional break, “Loop+breakZ”

3.2.3. Loop plus decrement, “Loop+dec”

3.3. Comments on the additional instructions

*3.3.1. Efficient simulation of other instructions with **breakZ***

3.3.2. “Breaks” and “exit”

*3.3.3. Efficiency independence of **breakZ** and **dec***

4. Lower bounds

4.1. Lower bounds based on the output size

4.2. Lower bounds for Loop – obstinacy-like results

4.2.1. Unary functions

4.2.2. General case

4.3. Loop+break and Loop+breakZ: lower bounds

4.3.1. Unary functions

4.3.2. Generalization

4.4. Loop+dec: lower bounds

5. Implementation and efficiency of some functions and constructs

5.1. Conditional instructions: implementation and efficiency

5.1.1. Conditional instructions: Loop language

5.1.2. Conditional instructions in Loop+break

5.1.3. Conjecture: not possible to simulate breakZ with break

5.1.4. Conditional instructions in Loop+dec

5.1.5. The “step” function

5.2. Decrement instruction: implementation and efficiency

5.3. Minimum function: implementation and efficiency

5.3.1. Minimum function in Loop

5.3.2. Minimum function in Loop+dec

5.3.3. Minimum function. Languages Loop+dec+breakZ and Loop+dec+break

5.4. Division by a constant

6. Conclusions

6.1. Main results

6.1.1. Comparison of the languages (efficiency)

6.1.2. Lower and upper bounds

6.2. Main conclusion

6.3. Further work