An analysis on the Invariance of λ-calculus with respect to Turing machines

Haileselassie Gaspar

May 23, 2025

1 Abstract

The notion of lambda calculus has been a part of computational theory ever since Alan Turing proved that it was an equivalent model to Turing machines, and it has had an significant impact on the field of computability and functional programming.

2 Introduction

3 Lambda Calculus

In order to talk about the invariance of λ-calculus it is first necesary to define some notation that will be used in this paper.

Definition. $M, N, P....$ denote arbitrary λ-terms, $x, y, z...$ denote variables and the set of λ-terms Λ is inductively defined as:

$$
\begin{aligned}
&\text{Variables: } x \in \Lambda \\
&\text{Abstraction: } M \in \Lambda(\lambda x.M) \in \Lambda \\
&\text{Application: } M, N \in \Lambda(MN) \in \Lambda
\end{aligned} \tag{1}
$$

Definition. $FV(M)$ is the set of free variables in M and it includes every variable in M not bound by an abstraction.

In order to analyze reduction strategies in the lambda calculus, we will introduce the concept of a context. A context is a lambda term with a parameter to be filled. It is defined as:

$$C ::= \langle \cdot \rangle \mid \lambda x.C \mid Ct \mid tC \tag{2}$$

For further reading on the syntax and axioms of the lambda calculus, refer to -Barendegt book-.

Definition. Let **R** be a notion of reduction on Λ. Then **R** induces the binary relations:

$$
\begin{aligned}
&1. \rightarrow_R \; \textit{one step R-reduction} \\
&2. \rightarrow_R^* \; \textit{R-reduction} \\
&3. =_R \; \textit{R-equality or R-convertibility}
\end{aligned}
\tag{3}
$$

4 Proof Overview

As stated before the measure employed to analyze the time invariance of lambda calculus is the number of transitions in a turing machine. By means of the Linear Substitution Calculus, it is possible to represent even size-exploding terms in Turing machines in polynomial time. It will be shown that by converting the LSC to a NPDA, the lambda calculus is indeed quadraticly bound in time when representing Turing machines.