Bachelor's Thesis Computing Science

Radboud University Nijmegen

Thesis Title

Subtitle if you like

Author:
Floris Reuvers
s1096976

First supervisor/assessor:
Dr. Niels van der Weide

Second assessor:
Dr. Engelbert Hubbers

April 9, 2025

Abstract

Brief outline of research questions, results. (The preferred size of an abstract is one paragraph or one page of text.)

Contents

Chapter 1

Introduction

The introduction of your bachelor thesis introduces the research area, the research hypothesis, and the scientific contributions of your work. A good narrative structure is the one suggested by Simon Peyton Jones [1]:

- describe the problem / research question
- motivate why this problem must be solved
- demonstrate that a (new) solution is needed
- explain the intuition behind your solution
- motivate why / how your solution solves the problem (this is technical)
- explain how it compares with related work

Close the introduction with a paragraph in which the content of the next chapters is briefly mentioned (one sentence per chapter).

Starting a new paragraph is done by inserting an empty line like this.

Chapter 2

Lambda Calculus

This section provides a detailed and formal description of the λ-calculus. We define a formal grammar of the λ-calculus and give examples of some terms in the λ-calculus. Then we explain β-reduction, after which we define call-by-name evaluation and call-by-value evaluation.

2.1 Introduction to the λ-calculus

The grammar of the lambda calculus is as follows:

$$M, N, P, Q \quad ::= \quad x \mid \lambda x.M \mid MN$$

Terms are denoted by M, N, P or Q and can either be of the form x, $\lambda x.M$ or MN:

- x is a variable, which is a symbol that represents an input.
- $\lambda x.M$ is an abstraction. An abstraction is an anonymous function, where x is the parameter and M is the body of the function.
- MN is a function application, where M and N are terms.

The λ-term $\lambda x.x$ is an abstraction. This specific abstraction is called the identity function and has one input parameter, namely x, and returns the input x. The body of the function is also x in this case and the function is often abbreviated as $\mathbf{I}$. The λ-term $(\lambda x.x)y$ is a function application. So the identity function is applied to the variable y and the λ-term reduces to the variable y.

2.2 β-Reduction

Before we discuss β-reduction, we first define bound and free variables and substitution.

2.2.1 Variables

Let us first define free variables. Let P be any term in the λ-calculus and $FV(P)$ be the set that contains all free variables in P. We define FV inductively on P. Since P can only be a variable, an abstraction or a function application, P can only be of the form x, $\lambda x.M$ and MN. Therefore, we define FV as follows:

$$\begin{aligned}
&FV(x) &&= \{x\}\\
&FV(\lambda x.M) &&= FV(M)\backslash\{x\}\\
&FV(MN) &&= FV(M) \cup FV(N)
\end{aligned}$$

The variable in a λ-term that solely consists of a variable is free. All free variables in an abstraction are those that are not the parameter of the abstraction. For example, the λ-term $\lambda x.yx$ has one free variable, y. We can use the following reasoning:

$$\begin{aligned}
FV(\lambda x.yx) &= FV(yx)\backslash\{x\}\\
&= (FV(x) \cup FV(y))\backslash\{x\}\\
&= (\{x\} \cup \{y\})\backslash\{x\}\\
&= \{x, y\}\backslash\{x\}\\
&= \{y\}
\end{aligned}$$

Let us now define bound variables. Again, let P be any term in the λ-calculus and $BV(P)$ be the set of all bound variables in P. We define BV inductively on P as follows:

$$\begin{aligned}
&BV(x) &&= \emptyset\\
&BV(\lambda x.M) &&= BV(M) \cup \{x\}\\
&BV(MN) &&= BV(M) \cup BV(N)
\end{aligned}$$

We say that the abstraction $\lambda x.M$ binds the variable x. Using our previous example of the λ-term $\lambda x.yx$, we can now reason that $BV(\lambda x.xy) = \{x\}$.

$$\begin{aligned}
BV(\lambda x.xy) &= BV(xy) \cup \{x\}\\
&= (BV(x) \cup BV(y)) \cup \{x\}\\
&= (\emptyset \cup \emptyset) \cup \{x\}\\
&= \{x\}
\end{aligned}$$

Let us consider a more tricky example: $\lambda x.(\lambda x.x)x$. Using the definitions of FV and BV, we can reason that $FV(\lambda x.(\lambda x.x)x) = \{\emptyset\}$ and $BV(\lambda x.(\lambda x.x)x) = \{x\}$. However, $FV(\lambda x.xx) = \{x\}$ and $BV(\lambda x.xx) =$

$\{x\}$, which means that the variable x is both free and bound. This may not be very intuitive, so we would prefer to write λ-term as $\lambda y.(\lambda x.x)y$ instead, where y is a new variable. These λ-terms are equivalent by the notion of *alpha equivalence*, which states that λ-terms that differ only by the names of bound variables are considered the same. We will discuss this in more detail after we define substitution.

2.2.2 Substitution

Keeping the definitions of free and bound variables in mind, we now define substitution. In this research, substitution is often denoted as $P[x \mapsto Q]$ and defined inductively on P by:

$$\begin{aligned}
x[x \mapsto Q] &= Q \\
y[x \mapsto Q] &= y \text{ if } (x \neq y) \\
(MN)[x \mapsto Q] &= M[x \mapsto Q]N[x \mapsto Q] \\
(\lambda x.M)[x \mapsto Q] &= \lambda x.M \\
(\lambda y.M)[x \mapsto Q] &= \lambda z.M[y \mapsto z][x \mapsto Q] \text{ if } (x \neq y)
\end{aligned}$$

with z a variable defined by:

1. If $x \notin FV(N)$ or $y \notin FV(M)$ then $z = y$
2. Otherwise, z can be any variable such that $z \notin FV(N)$ or $z \notin FV(M)$

The variable 'z' always exists. Choosing a fresh variable for z will always satisfy condition two. It might not be clear why need to use the variable z and one can expect that the following rule is good enough:

$$(\lambda y.M)[x \mapsto Q] = \lambda z.M[x \mapsto Q] \text{ if } (x \neq y)$$

However, the following example will illustrate why the more complicated rule is necessary. Consider the substitution $(\lambda y.xy)[x \mapsto y]$. The free variable of λ-term $\lambda y.xy$ is x and the bound variable is y. If we use the simple rule above, we get that $(\lambda y.xy)[x \mapsto y] = \lambda y.yy$, the meaning of the λ-term changes. However, the free variable we substituted, is not free anymore. The variable y is not a free variable in the term that we substituted y in. Therefore, we change the bound variable y in $\lambda y.xy$ to a new variable z. We can pick z as z is not a free variable in xy. Following the more complex rule, we get:

$$\begin{aligned}
(\lambda y.xy)[x \mapsto y] &= \lambda z.(xy)[y \mapsto z][x \mapsto y] \\
&= \lambda z.yz
\end{aligned}$$

2.2.3 α equivalence

Here, we define α equivalence. The idea is that renaming bound variables does not change the meaning of a λ-term. For example, the λ-terms $\lambda x.x$ and $\lambda y.y$ are equivalent ($\lambda x.x \equiv_a \lambda y.y$), since renaming all bound variables can result in identical λ-terms. The λ-terms $\lambda x.x$ and $\lambda y.y$ represent the same idea. We can define the relation $P \equiv_a P'$ inductively on P and P' as follows:

$$\begin{array}{ll} x \equiv_a y & \text{if } x = y \\ MN \equiv_a M'N' & \text{if } M \equiv_a M' \text{ and } N \equiv_a N' \\ \lambda x.M \equiv_a \lambda y.M' & \text{if } M \equiv_a M'[y \mapsto x] \text{ with } x = y \text{ or } x \notin FV(M') \end{array}$$

The last rule states that abstractions $\lambda x.M$ and $\lambda y.M'$ are equivalent if M and M' are equivalent after substituting y for x in M'. However, x should be equal to y or x should not be a free variable in M'.

- $\lambda x.zx$ is equivalent to $\lambda y.zy$, since $zy[y \mapsto x] = zx \equiv_a zx$ and x is not a free variable in zy.
- $\lambda z.zx$ is not to $\lambda y.zy$, since z is a free variable in zy and x is not equal to y.

2.2.4 β-Reduction

Now, we are ready to discuss β-reduction. In the λ-calculus, there is one way to simplify terms, which is β-reduction. We can define:

$$(\lambda x.M)N \to M[x \mapsto N] \qquad (\beta)$$

So a λ-term of the form $(\lambda x.M)N$ can be reduced by substituting x by N in M. However, without additional rules, we cannot apply β-reduction to subterms. For example, the following reduction would not be possible: $\lambda x.(\lambda y.y)x \to \lambda x.x$. Therefore, we can define the following rules:

$$\frac{MN \to M'N}{M \to M'}\,(\mu) \qquad \frac{MN \to MN'}{N \to N'}\,(\nu) \qquad \frac{\lambda x.M \to \lambda x.M'}{M \to M'}\,(\xi)$$

Now, we define $\to_\beta$ as β closed under μ, ν and ξ. So with $\to_\beta$ we can use β-reduction on all subterms if the subterm is of the form $(\lambda x.M)N$. A subterm of the form $(\lambda x.M)N$ is called a β-redex. In one reduction, we may need to use multiple rules. The β-redex that is reduced is underlined. For instance, we need to use the μ and ξ rule in the last example of the following reductions:

	β	μ	ν	$\nu_<$	ξ
$\rightarrow_w$	β_n	✓	✓		
$\rightarrow_n$	β_n	✓			

$(\underline{\mathbf{II}})(\mathbf{II}) \rightarrow_\beta \mathbf{I}(\mathbf{II})$	β with μ rule
$(\mathbf{II})(\underline{\mathbf{II}}) \rightarrow_\beta (\mathbf{II})\mathbf{I}$	β with ν rule
$\lambda x.\underline{\mathbf{I}x} \rightarrow_\beta \lambda x.x$	β with ξ rule
$(\mathbf{I}(\lambda x.\underline{\mathbf{I}x}))(\mathbf{II}) \rightarrow_\beta (\mathbf{I}(\lambda x.x))(\mathbf{II})$	β with μ and ξ rule

2.3 Call-by-name calculus

In this section, we discuss the call-by-name λ-calculus. The reduction rule that is used in this calculus is the same as the β reduction rule. However, we name the rule β_n to make it clear that it is the reduction rule of the call-by-name λ-calculus.

$$(\lambda x.M)N \rightarrow M[x \mapsto N] \qquad \beta_n$$

We can define weak reduction, $\rightarrow_w$, as β_n closed under μ and ν. So $\rightarrow_w$ has the only restiction that it cannot reduce under λ's. The relation $\rightarrow_n$ is defined as β_n closed under μ. So using name evaluation, we can not evaluate the argument of a function. Call-by-name evaluation is defined as $\rightarrow_n^*$. In the following example, we give the call-by-name evaluation of the λ-term $(\mathbf{I}(\lambda x.\mathbf{I}x))(\mathbf{II})$. The β-redex that is reduced is underlined. Note that $\rightarrow_n$ is deterministic, so there is only one way to apply $\rightarrow_n$ on a λ-term.

$$\begin{array}{rcl}
(\underline{\mathbf{I}(\lambda x.\mathbf{I}x)})(\mathbf{II}) & \rightarrow_n & \underline{(\lambda x.\mathbf{I}x)(\mathbf{II})} \\
 & \rightarrow_n & \underline{\mathbf{I}(\mathbf{II})} \\
 & \rightarrow_n & \underline{\mathbf{II}} \\
 & \rightarrow_n & \mathbf{I}
\end{array}$$

2.4 Call-by-value calculus

Before we can define the call-by-value λ-calculus (abbreviated as), we first define what we mean by values. Values are all λ-terms that are either a variable or an abstractions. Values are usually denoted by V or W.

Closure Rules				
	β	μ	ν	ξ
$\rightarrow_{\beta_n}$	β_n	✓	✓	✓
$\rightarrow_n$	β_n	✓		
$\rightarrow_w$	β_n	✓	✓	

$$V \quad ::= \quad x \mid \lambda x.M$$

In the λ_v, we can only reduce if the argument is a value. Therefore, the reduction rule is as follows:

$$(\lambda x.M)V \rightarrow M[x \mapsto V] \qquad \beta_v$$

In order to define the $\rightarrow_v$ relation, we first define the rule $\nu_<$.

$$\frac{N \rightarrow N'}{VN \rightarrow VN'}\,\nu_<$$

The relation $\rightarrow_v$ can be defined as β_v, closed under μ and $\nu_<$. The $\nu_<$ forces evaluation from left to right and β_v makes sure that the argument needs to be a value. In the following example, we give the call-by-value evaluation of the λ-term $(\mathbf{I}(\lambda x.\mathbf{I}x))(\mathbf{II})$. The β-redex that is reduced is underlined. Note that $\rightarrow_v$ is deterministic, so there is only one way to apply $\rightarrow_v$ on a λ-term.

$$\begin{array}{rcl}
(\underline{\mathbf{I}(\lambda x.\mathbf{I}x)})(\mathbf{II}) & \rightarrow_v & (\lambda x.\mathbf{I}x)(\underline{\mathbf{II}}) \\
 & \rightarrow_v & \underline{(\lambda x.\mathbf{I}x)\mathbf{I}} \\
 & \rightarrow_v & \underline{\mathbf{II}} \\
 & \rightarrow_v & \mathbf{I}
\end{array}$$

Closure Rules					
	β	μ	ν	$\nu_<$	ξ
$\rightarrow_{\beta_n}$	β_n	✓	✓	✓	✓
$\rightarrow_n$	β_n	✓			
$\rightarrow_w$	β_n	✓	✓		
$\rightarrow_{\beta_v}$	β_v	✓	✓	✓	✓
$\rightarrow_v$	β_v	✓		✓	

Chapter 3

Rules

3.1 Rules Intuitionistic Logic

$$\frac{}{A \vdash A}\,\text{Id} \qquad \frac{\Gamma, \Delta \vdash A}{\Delta, \Gamma \vdash A}\,\text{Exchange}$$

$$\frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\,\text{Contraction} \qquad \frac{\Gamma \vdash B}{\Gamma, A \vdash B}\,\text{Weakening}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,\to\text{-I} \qquad \frac{\Gamma \vdash A \to B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B}\,\to\text{-E}$$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \times B}\,\times\text{-I} \qquad \frac{\Gamma \vdash A \times B \qquad \Delta, A, B \vdash C}{\Gamma, \Delta \vdash C}\,\times\text{-E}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A + B}\,+\text{-I}_1 \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A + B}\,+\text{-I}_2$$

$$\frac{\Gamma, A \vdash C \qquad \Delta, A \vdash C \qquad \Delta, B \vdash C}{\Gamma, \Delta \vdash C}\,+\text{-E}$$

Figure 3.1: Multiple Proof Rules in Columns

3.2 Rules Intuitionistic Logic with Types

$$\frac{}{x : A \vdash x : A}\,\text{Id} \qquad \frac{\Gamma, \Delta \vdash t : A}{\Delta, \Gamma \vdash t : A}\,\text{Exchange}$$

$$\frac{\Gamma, y : A, z : A \vdash u : B}{\Gamma, x : A \vdash u[y \mapsto x][z \mapsto x] : B}\,\text{Contraction}$$

$$\frac{\Gamma \vdash u : B}{\Gamma, x : A \vdash u : B}\,\text{Weakening}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\,\to\text{-I}$$

$$\frac{\Gamma \vdash s : A \to B \qquad \Delta \vdash t : A}{\Gamma, \Delta \vdash s(t) : B}\,\to\text{-E}$$

$$\frac{\Gamma \vdash t : A \qquad \Delta \vdash u : B}{\Gamma, \Delta \vdash (t, u) : A \times B}\,\times\text{-I}$$

$$\frac{\Gamma \vdash s : A \times B \qquad \Delta, x : A, y : B \vdash v : C}{\Gamma, \Delta \vdash \text{case } s \text{ of } (x, y) \to v : C}\,\times\text{-E}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \text{inl}(t) : A + B}\,+\text{-I}_1 \qquad \frac{\Gamma \vdash u : B}{\Gamma \vdash \text{inr}(u) : A + B}\,+\text{-I}_2$$

$$\frac{\Gamma \vdash s : A + B \qquad \Delta, x : A \vdash v : C \qquad \Delta, y : B \vdash w : C}{\Gamma, \Delta \vdash \text{case } s \text{ of inl}(x) \to v;\ \text{inr}(y) \to w : C}\,+\text{-E}$$

Figure 3.2: Multiple Proof Rules in Columns

3.3 Rules Linear Logic

$$\frac{}{\langle A\rangle \vdash A}\langle\mathrm{Id}\rangle \qquad \frac{}{[A] \vdash A}[\mathrm{Id}] \qquad \frac{\Gamma, \Delta \vdash A}{\Delta, \Gamma \vdash A}\text{Exchange}$$

$$\frac{\Gamma, [A], [A] \vdash B}{\Gamma, [A] \vdash B}\text{Contraction} \qquad \frac{\Gamma \vdash B}{\Gamma, [A] \vdash B}\text{Weakening}$$

$$\frac{[\Gamma] \vdash A}{[\Gamma] \vdash !A}\text{!-I} \qquad \frac{\Gamma \vdash !A \qquad \Delta, [A] \vdash B}{\Gamma, \Delta \vdash B}\text{!-E}$$

$$\frac{\Gamma, \langle A\rangle \vdash B}{\Gamma \vdash A \multimap B}\multimap\text{-I} \qquad \frac{\Gamma \vdash A \multimap B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B}\multimap\text{-E}$$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\otimes\text{-I}$$

$$\frac{\Gamma \vdash A \otimes B \qquad \Delta, \langle A\rangle, \langle B\rangle \vdash C}{\Gamma, \Delta \vdash C}\otimes\text{-E}$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \,\&\, B}\&\text{-I}$$

$$\frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash A}\&\text{-E}_1 \qquad \frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash B}\&\text{-E}_2$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B}\oplus\text{-I}_1 \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B}\oplus\text{-I}_2$$

$$\frac{\Gamma \vdash A \oplus B \qquad \Delta, \langle A\rangle \vdash C \qquad \Delta, \langle B\rangle \vdash C}{\Gamma, \Delta \vdash C}\oplus\text{-E}$$

Figure 3.3: Multiple Proof Rules in Columns

3.4 Rules Linear Logic with Types

$$\frac{}{\langle x : A\rangle \vdash x : A}\ \langle\text{Id}\rangle \qquad \frac{}{[x : A] \vdash x : A}\ [\text{Id}]$$

$$\frac{\Gamma, \Delta \vdash t : A}{\Delta, \Gamma \vdash t : A}\ \text{Exchange} \qquad \frac{\Gamma \vdash u : B}{\Gamma, [x : A] \vdash u : B}\ \text{Weakening}$$

$$\frac{\Gamma, [y : A], [z : A] \vdash u : B}{\Gamma, [x : A] \vdash u[y \mapsto x][z \mapsto x] : B}\ \text{Contraction}$$

$$\frac{[\Gamma] \vdash t : A}{[\Gamma] \vdash !t : !A}\ \text{!-I}$$

$$\frac{\Gamma \vdash s : !A \qquad \Delta, [x : A] \vdash u : B}{\Gamma, \Delta \vdash \text{case } s \text{ of } !x \to u : B}\ \text{!-E}$$

$$\frac{\Gamma, \langle x : A\rangle \vdash u : B}{\Gamma \vdash \lambda\langle x\rangle.u : A \multimap B}\ \multimap\text{-I}$$

$$\frac{\Gamma \vdash s : A \multimap B \qquad \Delta \vdash t : A}{\Gamma, \Delta \vdash s\langle t\rangle : B}\ \multimap\text{-E}$$

$$\frac{\Gamma \vdash t : A \qquad \Delta \vdash u : B}{\Gamma, \Delta \vdash \langle t, u\rangle : A \otimes B}\ \otimes\text{-I}$$

$$\frac{\Gamma \vdash s : A \otimes B \qquad \Delta, \langle x : A\rangle, \langle y : B\rangle \vdash v : C}{\Gamma, \Delta \vdash \text{case } s \text{ of } \langle x, y\rangle \to v : C}\ \otimes\text{-E}$$

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : B}{\Gamma \vdash \langle\langle t, u\rangle\rangle : A \,\&\, B}\ \&\text{-I}$$

$$\frac{\Gamma \vdash s : A \,\&\, B}{\Gamma \vdash \text{fst}\langle s\rangle : A}\ \&\text{-E}_1 \qquad \frac{\Gamma \vdash s : A \,\&\, B}{\Gamma \vdash \text{snd}\langle s\rangle : B}\ \&\text{-E}_2$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \text{inl}\langle t\rangle : A \oplus B}\ \oplus\text{-I}_1 \qquad \frac{\Gamma \vdash u : B}{\Gamma \vdash \text{inr}\langle u\rangle : A \oplus B}\ \oplus\text{-I}_2$$

$$\frac{\Gamma \vdash s : A \oplus B \qquad \Delta, \langle x : A\rangle \vdash v : C \qquad \Delta, \langle y : B\rangle \vdash w : C}{\Gamma, \Delta \vdash \text{case } s \text{ of inl}\langle x\rangle \to v;\ \text{inr}\langle y\rangle \to w : C}\ \oplus\text{-E}$$

Figure 3.4: Multiple Proof Rules in Columns

Chapter 4

Call-by-box Lambda Calculus

4.1 Speficitation of $\boldsymbol{\lambda_b}$

The terms of λ_b are given by:
$M, N, P, Q \quad ::= \quad \varepsilon(x) \mid \lambda x.M \mid MN \mid \text{box}(N)$
Types are given by:
$A, B \quad ::= \quad X \mid A \to B \mid \Box A$
The typing rules of λ_b are give by:

$$\frac{}{\Gamma, x : \Box A \vdash \varepsilon(x) : A}\ \text{Id}_\Box$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}\ \to\text{-I}$$

$$\frac{\Gamma \vdash M : A \to B \qquad \Delta \vdash N : A}{\Gamma, \Delta \vdash MN : B}\ \to\text{-E}$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \text{box}(M) : \Box A}\ \Box\text{-I}$$

Since variables always occur inside ε, we need to redefine free and bound variables and substitution. For λ_b, we have:

$$\begin{array}{llll} FV_b(\varepsilon(x)) & = \{\varepsilon(x)\} & BV(\varepsilon(x)) & = \emptyset \\ FV_b(\lambda x.M) & = FV(M) \backslash \{\varepsilon(x)\} & BV(\lambda x.M) & = BV(M) \cup \{\varepsilon(x)\} \\ FV_b(MN) & = FV(M) \cup FV(N) & BV(MN) & = BV(M) \cup BV(N) \end{array}$$

Closure Rules						
	β	μ	$\mu_>$	ν	$\nu_<$	ξ
$\rightarrow_{\beta_n}$	β_n	✓	-	✓	✓	✓
$\rightarrow_n$	β_n	✓	-			
$\rightarrow_w$	β_n	✓	-	✓		
$\rightarrow_{\beta_v}$	β_v	✓	-	✓	✓	✓
$\rightarrow_v$	β_v	✓	-		✓	
$\rightarrow_{\beta_b}$	β_b	✓	✓	✓	✓	✓
$\rightarrow_{we}$	β_b	✓	✓	✓		
$\rightarrow_{we_<}$	β_b		✓	✓		

Thus, for substitution, we get:

$$\begin{aligned}
&\varepsilon(x)[\varepsilon(x) \mapsto Q] && = Q \\
&\varepsilon(y)[\varepsilon(x) \mapsto Q] && = \varepsilon(y) \text{ if } (x \neq y) \\
&(MN)[\varepsilon(x) \mapsto Q] && = M[x \mapsto Q]N[x \mapsto Q] \\
&(\lambda x.M)[\varepsilon(x) \mapsto Q] && = \lambda x.M \\
&(\lambda y.M)[\varepsilon(x) \mapsto Q] && = \lambda z.M[\varepsilon(y) \mapsto \varepsilon(z)][\varepsilon(x) \mapsto Q] \text{ if } (x \neq y)
\end{aligned}$$

with z a variable defined by:

1. If $\varepsilon(x) \notin FV_b(N)$ or $\varepsilon(y) \notin FV_b(M)$ then $z = y$
2. Otherwise, z can be any variable such that $\varepsilon(z) \notin FV(N)$ or $\varepsilon(z) \notin FV(M)$

The reduction rule we can use for this λ-calculus is:

$$(\lambda x.M)\text{box}(N) \rightarrow M[\varepsilon(x) \mapsto N] \qquad (\beta_b)$$

We define two relations: $\rightarrow_{b_n}$ and $\rightarrow_{b_v}$. The relation $\rightarrow_{b_n}$ will be used for cbn evaluation for embeddings of λ_n in λ_b. The relation $\rightarrow_{b_v}$ will be used for cbn evalluation for embeddings of λ_v in λ_b.

4.2 Embedding of $\boldsymbol{\lambda_n}$ into $\boldsymbol{\lambda_b}$

Let Λ_{CBN} be the set containing all λ-terms in the λ_n and let Λ_{CBB} be the set containing all λ-terms in the λ_b. The set $\mathcal{T}_{CBN}$ contains all types of the λ_n and $\mathcal{T}_{CBB}$ contains all types of the λ_b. Now we define functions from Λ_{CBN} to Λ_{CBB} for term translation and from $\mathcal{T}_{CBN}$ to $\mathcal{T}_{CBB}$ for type translation. These functions represent an embedding of the λ_n into the λ_b.

$$\begin{array}{llll}
T_t & : \mathcal{T}_{CBN} \to \mathcal{T}_{CBB} & T & : \Lambda_{CBN} \to \Lambda_{CBB} \\
T_t(X) & = X & T(x) & = \varepsilon(x) \\
T_t(A \to B) & = \Box T_t(A) \to T_t(B) & T(\lambda x.M) & = \lambda x.T(M) \\
 & & T(MN) & = T(M)\text{box}(T(N))
\end{array}$$

To clarify, we give the embedding of $\mathbf{I}x$ into λ_b after which we apply $\to_{b_n}$. It is clear that the result obtained after $\to_{b_n}$ is equal to the embedding of the result of applying $\to_{b_n}$ on $\mathbf{I}x$ in the λ_n, which is $T(x) = \varepsilon(x)$.

$$\begin{array}{rcl}
T(\mathbf{I}x) & = & (T(\lambda x.x))\text{box}(T(x)) \\
 & = & (\lambda x.T(x))\text{box}(T(x)) \\
 & = & (\lambda x.\varepsilon(x))\text{box}(\varepsilon(x)) \\
 & \to_{b_n} & (\lambda x.\varepsilon(x))[\varepsilon(x) \mapsto \varepsilon(x)] \\
 & = & \varepsilon(x)
\end{array}$$

4.3 Embedding of $\boldsymbol{\lambda_v}$ into $\boldsymbol{\lambda_b}$

For the embedding of the λ_v into the λ_b, we make use of the abbreviation ”raise” defined by:

$$\text{raise}(M) := \lambda z.\varepsilon(z)M$$

Let Λ_{CBV} be the set containing all λ-terms in the λ_v and let Λ_{CBB} be the set containing all λ-terms in the λ_b. The set $\mathcal{T}_{CBV}$ contains all types of the λ_v and $\mathcal{T}_{CBB}$ contains all types of the λ_b. Now we define functions from Λ_{CBV} to Λ_{CBB} for term translation and from $\mathcal{T}_{CBV}$ to $\mathcal{T}_{CBB}$ for type translation.

$$\begin{array}{llll}
T_t & : \mathcal{T}_{CBV} \to \mathcal{T}_{CBB} & T & : \Lambda_{CBN} \to \Lambda_{CBB} \\
T_t(X) & = \Box X & T(x) & = \text{box}(\varepsilon(x)) \\
T_t(A \to B) & = \Box(\Box T_t(A) \to \Box T_t(B)) & T(\lambda x.M) & = \text{box}(\lambda x.T(M)) \\
 & & T(MN) & = \text{raise}(T(N))T(M)
\end{array}$$

To clarify, we give the embedding of $\mathbf{I}x$ into λ_b after which we apply $\to_{b_v}$ twice. It is clear that the result obtained after $\to_{b_v}$ is equal to the embedding of the result of applying $\to_v$ on $\mathbf{I}x$ in the λ_v, which is $T(x) = \text{box}(\varepsilon(x))$.

$$\begin{aligned}
T(\mathbf{I}x) &= \text{raise}(T(x))T(\mathbf{I}) \\
&= \text{raise}(T(x))T(\lambda x.x) \\
&= \text{raise}(\text{box}(\varepsilon(x)))\text{box}(\lambda x.T(x)) \\
&= \text{raise}(\text{box}(\varepsilon(x)))\text{box}(\lambda x.\text{box}(\varepsilon(x))) \\
&= (\lambda z.\varepsilon(z)\text{box}(\varepsilon(x)))\text{box}(\lambda x.\text{box}(\varepsilon(x))) \\
&\to_{b_v} (\lambda x.\text{box}(\varepsilon(x)))\text{box}(\varepsilon(x)) \\
&\to_{b_v} \text{box}(\varepsilon(x))
\end{aligned}$$

Chapter 5

Formalisation in Agda

5.1 Box Calculus

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

5.2 Linear Logic

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec tincidunt, nunc ut bibendum facilisis, nisi nunc fringilla nisi, nec mollis nunc nisi nec ligula. Donec id ligula ac enim efficitur mollis. Donec id ligula ac enim efficitur mollis. Donec id ligula ac enim efficitur mollis. Donec id ligula ac enim efficitur mollis.

Chapter 6

Related Work

In this chapter you demonstrate that you are sufficiently aware of the state-of-art knowledge of the problem domain that you have investigated as well as demonstrating that you have found a *new* solution / approach / method.

Chapter 7

Conclusions

In this chapter you present all conclusions that can be drawn from the preceding chapters. It should not introduce new experiments, theories, investigations, etc.: these should have been written down earlier in the thesis. Therefore, conclusions can be brief and to the point.

Bibliography

[1] Simon Peyton Jones. How to write a good research paper, 2004. Presentation at Technical University of Vienna, `http://research.microsoft.com/en-us/um/people/simonpj/papers/giving-a-talk/writing-a-paper-slides.pdf`.

Appendix A

Appendix

Appendices are *optional* chapters in which you cover additional material that is required to support your hypothesis, experiments, measurements, conclusions, etc. that would otherwise clutter the presentation of your research.