A Report of *Type Theory and Formal Proof*

Juan Pablo Royo Sales

Universitat Politècnica de Catalunya

February 10, 2021

Contents

1 Introduction

This report is going to provide a summary over the book [NG14]. Alongside the different chapters of the book I am going to describe briefly the most important parts of each chapter and, at the same time, I am going to solve 1 or 2 of the exercises proposed by the authors.

The organization of the report is going to be the same as the chapters of the book.

2 Untyped lambda calculus

In this first chapter the authors define and describe Lambda Calculus (λ-*calculus*) system which encapsulates the formalization of basic aspects of mathematical functions, in particular construction and use. In λ-*calculus* formalization system there are *typed* and *untyped* formalization of the same system. In this first case authors introduced the first basic and simple formalization which is *untyped.*

2.1 Definition

There are *two constructions principles* and *one evaluation rule*

Construction principles:

- *Abstraction:* Given an expression M and a variable x we can construct the expression: $\lambda x.M$. This is abstraction of x over M Example: $\lambda y.(\lambda x.x - y)$ Abstraction of y over $\lambda x.x - y$
- *Application:* Given 2 expressions M and N we can construct the expression: $M\ N$. This is the application of M to N. Example: $(\lambda x.x^2+1)(3)$ Application of 3 over $\lambda x.x^2 + 1$

Evaluation Rule: Formalization of this process is called Beta Reduction (β-*reduction*). β-*reduction*: An expression $(\lambda x.M)N$ can be rewritten to $M[x := N]$, which means every x should be replaced by N in M. This process is called β-*reduction* of $(\lambda x.M)N$ to $M[x := N]$.

Example: $(\lambda x.x^2 + 1)(3)$ reduces to $(x^2 + 1)[x := 3]$, which is $3^2 + 1$.

In this book, functions on λ-*calculus* notation are *Curried.*

2.1.1 Lambda-terms

Expressions in λ-*calculus* are called Lambda Terms (λ-*term*)

Definition 1. *The set Λ of all λ-term*

1. (Variable) If $u \in V$, then $u \in \Lambda$
   Example: x, y, z
2. (Application) If M and $N \in \Lambda$, then $(MN) \in \Lambda$
   Example: (xy), $(x(xy))$
3. (Abstraction) If $u \in V$ and $M \in \Lambda$, then $(\lambda u.M) \in \Lambda$
   Example: $(\lambda x.(xz))$, $(\lambda y.(\lambda z.x))$

Definition 2. *Multiset of subterms Sub*

1. (Basis) $Sub(x) = \{x\}$, for each $x \in V$
2. (Application) $Sub((MN)) = Sub(M) \cup Sub(N) \cup \{(MN)\}$
3. (Abstraction) $Sub((\lambda x.M)) = Sub(M) \cup \{(\lambda x.M)\}$

Lemma 1. *(1) (Reflexivity) For all λ-term M, we have $M \in Sub(M)$. (2) (Transitivity) If $L \in Sub(M)$ and $M \in Sub(N)$, then $L \in Sub(N)$.*

Definition 3 (Proper subterm)**.** *L is a proper subterm of M if L is a subterm of M, but $L \not\equiv M$*

- Parenthesis can be omitted
- Application is lef-associative, MNL is $((MN)L)$
- Application takes precedence over Abstraction

2.2 Free and bound variables

Variables can be *free*, *bound* and *binding*. A variable x which is *free* in M becomes *bound* in $\lambda x.M$. M is called a *binding* variable occurrence.

Definition 4 (FV, set of free variables of a λ-*term*)**.**

1. (Variable) $FV(x) = \{x\}$
2. (Application) $FV(MN) = FV(M) \cup FV(N)$
3. (Abstraction) $FV(\lambda x.M) = FV(M) \setminus \{x\}$

Definition 5 (Closed λ-*term*; combinator; Λ^0)**.** *The λ-term M is closed if $FV(M) = \emptyset$. This is also called a combinator. The set of all closed λ-term is denoted by Λ^0*

2.2.1 Alpha conversion

It is based on the possibility of renaming bound and binding variables.

Definition 6 (Renaming; $M^{x\to y}$; $=_\alpha$)**.** *Let $M^{x\to y}$ denote the result of replacing every free ocurrence of x in M by y. Renaming, expressed by $=_\alpha$ is defined as: $\lambda x.M =_\alpha \lambda y.M^{x\to y}$, provided that $y \notin FV(M)$ and y is not binding in M*

Definition 7 (α-convertion or α-equivalence; $=_\alpha$)**.**

1. (Renaming) same as 6

2. (Compatibility) If $M =_\alpha N$, then $ML =_\alpha NL$, $LM =_\alpha LN$ and, for any arbitrary z, $\lambda z.M =_\alpha \lambda z.N$
3. (Reflexivity) $M =_\alpha M$
4. (Symmetry) If $M =_\alpha N$ then $N =_\alpha M$
5. (Transitivity) If both $L =_\alpha M$ and $M =_\alpha N$, then $L =_\alpha N$

2.3 Substitution

Definition 8 (Substitution)**.**

1. $x[x := N] \equiv N$
2. $y[x := N] \equiv y$ if $x \not\equiv y$
3. $(PQ)[x := N] \equiv (P[x := N])(Q[x := N])$
4. $(\lambda y.P)[x := N] \equiv \lambda z.(P^{y\to z}[x := N])$, if $\lambda z.P^{y\to z}$ is α-variant of $\lambda y.P$ such that $z \notin FV(N)$

2.4 Beta reduction

Definition 9 (One-step β-reduction, $\to_\beta$)**.**

1. (Basis) $(\lambda x.M)N \to_\beta M[x := N]$,
2. (Compatibility) If $M \to_\beta N$, then $ML \to_\beta NL$, $LM \to_\beta LN$ and $\lambda x.M \to_\beta \lambda x.N$

In 1 the left part of $\to_\beta$ is called *redex* (reducible expression), and the right side is called *contractum* (of the redex).

Definition 10 (β-reduction (zero-or-more-step), $\twoheadrightarrow \beta$)**.** *$M \twoheadrightarrow_\beta N$ if there is an $n \geq 0$ and there are terms M_0 to M_n such that $M_0 \equiv M$, $M_n \equiv N$ and for all $i, 0 \leq i < n$:*
$M_i \to_\beta M_{i+1}$

Hence, if $M \twoheadrightarrow_\beta N$, there exists a chain of single-step β-reductions, starting with M and ending with N:

$M \equiv M_0 \to_\beta M_1 \to_\beta M_2 \to_\beta \cdots \to_\beta M_{n-2} \to_\beta M_{n-1} \to_\beta M_n \equiv N$

Definition 11 (β-conversion, β-equality; $=_\beta$)**.** *$M =_\beta N$ if there is an $n \geq 0$ and there are terms M_0 to M_n such that $M_0 \equiv M$, $M_n \equiv N$ and for all $i, 0 \leq i < n$:*
either $M_i \to_\beta M_{i+1}$ or $M_{i+1} \to_\beta M_i$

2.5 Fixed Point Theorem

Theorem 1. *For all $L \in \Lambda$ there is $M \in \Lambda$ such that $LM =_\beta M$*

Proof. For given L, define $M := (\lambda x.L(xx))(\lambda x.L(xx))$ This M is a redex, so we have:

$$
\begin{aligned}
M &\equiv (\lambda x.L(xx))(\lambda x.L(xx)) && \text{(1a)}\\
&\to_\beta L((\lambda x.L(xx))(\lambda x.L(xx))) && \text{(1b)}\\
&\equiv LM && \text{(1c)}
\end{aligned}
$$

Therefore, $LM =_\beta M$ □

References

[NG14] Rob Nederpelt and Herman Geuvers. *Type Theory and Formal Proof.* Cambridge University Press, Cambridge CB2 8BS, United Kindom, 2014.