

Type Theory Journal

Amir H. Ebrahimnezhad

April 11, 2024

Contents

1 Simple Type Theory	1
1.1 Syntax	1

1 Simple Type Theory

1.1 Syntax

There are two notations for types and expressions, a formal notation based on a small number of constructors and a compact notation that resembles the notation found in mathematical practice. The formal notation is an internal syntax intended for machines. The compact notation is introduced using notational definitions and notational conventions, intended for human.

A notational definition has the form A stands for B . Where A and B are notations that represents types or expressions, if defines A to be a more compact, convenient, or standard notation for presenting the type or expression that B represents.

Definition 1 –Let S_{bt} , S_{var} , and S_{con} be fixed countably infinite sets of symbols that will serve as names of base types, variables, and constants, respectively.

Definition 2 –A **type** of Alonzo is an inductive set of strings of symbols defined by the following instructions

- *Type of truth values*: BoolTy is a type.
- *Base type*: $\text{BaseTy}(a)$ is a type.
- *Function Type*: $\text{FunTy}(\alpha, \beta)$ is a type.
- *Product Type*: $\text{ProdTy}(\alpha, \beta)$ is a type.

Hence, a type is a string constructed from the members of S_{bt} via the constructors BoolTy , $\text{BaseTy}(a)$, $\text{FunTy}(\alpha, \beta)$, $\text{ProdTy}(\alpha, \beta)$. A type is presented in the formal notation when it is written as a string according to the definition above. However, we will usually present types using the compact notation defined as below. We might also omit parentheses when ever there's no loss of meaning.

- o stands for BoolTy .
- a stands for $\text{BaseTy}(a)$.

- $(\alpha \rightarrow \beta)$ stands for $\text{FunTy}(\alpha, \beta)$.
- $(\alpha \times \beta)$ stands for $\text{ProdTy}(\alpha, \beta)$.

Definition 3 –An **Expression of type α** of Alonzo is an inductive set of strings of symbols defined by the following constructors.

- *Variable*: $\text{Var}(\mathbf{x}, \alpha)$ is an expression of type α .
- *Constant*: $\text{Con}(\mathbf{c}, \alpha)$ is an expression of type α .
- *Equality*: $\text{Eq}(\mathbf{A}_\alpha, \mathbf{B}_\alpha)$ is an expression of type α .
- *Function Application*: $\text{FunApp}(\mathbf{F}_{\text{FunTy}(\alpha, \beta)}, \mathbf{A}_\alpha)$ is an expression of type β .
- *Function Abstraction*: $\text{FunAbs}(\text{Var}(\mathbf{x}, \alpha), \mathbf{B}_\beta)$ is an expression of type $\text{FunTy}(\alpha, \beta)$.
- *Definite Description*: $\text{DefDes}(\text{Var}(\mathbf{x}, \alpha), \mathbf{A}_o)$ is an expression of type α where $\alpha \neq \text{BoolTy}$.
- *Ordered Pair*: $\text{OrdPair}(\mathbf{A}_\alpha, \mathbf{B}_\beta)$ is an expression of type $\text{ProdTy}(\alpha, \beta)$.

Again we define our compact notation of the formal notation above:

- $\mathbf{x} : \alpha$ stands for $\text{Var}(\mathbf{x}, \alpha)$.
- \mathbf{c}_α stands for $\text{Con}(\mathbf{c}, \alpha)$.
- $\mathbf{A}_\alpha = \mathbf{B}_\alpha$ stands for $\text{Eq}(\mathbf{A}_\alpha, \mathbf{B}_\alpha)$.
- $\mathbf{F}_{\alpha \rightarrow \beta} \mathbf{A}_\alpha$ stands for $\text{FunApp}(\mathbf{F}_{\alpha \rightarrow \beta}, \mathbf{A}_\alpha)$.
- $\lambda \mathbf{x} : \alpha. \mathbf{B}_\beta$ stands for $\text{FunAbs}(\text{Var}(\mathbf{x}, \alpha), \mathbf{B}_\beta)$.
- $\mathbf{I} \mathbf{x} : \alpha. \mathbf{A}_o$ stands for $\text{DefDes}(\text{Var}(\mathbf{x}, \alpha), \mathbf{A}_o)$.

Callout — $\mathbf{A}_\alpha \equiv \mathbf{B}_\alpha$ means that the expressions denoted by \mathbf{A}_α and \mathbf{B}_α , is identical.

Definition 4 –An occurrence of a variable $\mathbf{x} : \alpha$ is **bound** if it is within a subexpression of \mathbf{B}_β either the form $\lambda \mathbf{x} : \alpha. \mathbf{C}_\gamma$ or the form $\mathbf{I} \mathbf{x} : \alpha. \mathbf{C}_o$

Definition 5 –An occurrence of a variable $\mathbf{x} : \alpha$ is **free** if it is not within a subexpression of \mathbf{B}_β either the form $\lambda \mathbf{x} : \alpha. \mathbf{C}_\gamma$ or the form $\mathbf{I} \mathbf{x} : \alpha. \mathbf{C}_o$

References