

Lean

PENG

Created: 20260910, Version: 20260910

Contents

1	Introduction	1
1.1	Evaluating Expressions	1
1.2	Types	1
1.3	Functions and Definitions	1
1.3.1	Defining Functions	2
1.3.2	Defining Types	2

1 Introduction

Lean is a strict pure functional language with dependent types [1].

1.1 Evaluating Expressions

1.2 Types

Every expression in Lean must have a type before it can be evaluated.

Definition 1.1 (type).

- `Nat`: *default, natural numbers* ≥ 0
- `Int`
- `String`
- `Bool`

Example 1.2.

```
#eval (1 - 2 : Nat) -- 0
#check (1 - 2 : Int) -- check the type of an expression
```

1.3 Functions and Definitions

Definition 1.3 (definition).

```
def var : type := value
```

Example 1.4.

```
def hello := "Hello"
def lean : String := "Lean"
#eval String.append hello (String.append " " lean)
```

1.3.1 Defining Functions

Definition 1.5 (function).

```
def func (arg1 : type) ... (argn : type) : type := body
```

Example 1.6.

```
def maximum (n : Nat) (m : Nat) : Nat :=  
if n < m then  
  m  
else n  
#eval maximum 2 3  
#check maximum
```

Remark 1.7. *All functions actually expect precisely one argument. Functions to take more than one arguments are in fact functions that take one argument and then return a new function.*

1.3.2 Defining Types

References

- [1] David Thrane Christiansen. *Functional programming in Lean*. 2023.