A Simple Python Formalisation of ‘Book HoTT’

Clarence Protin

April 8, 2023

Abstract

In the programming style of [1] we present a simple minimal typing assistant for ‘Book HoTT’, the second presentation of Dependent Type Theory and the Univalence Axiom in the online book Homotopy Type Theory: Univalent Foundations of Mathematics, `https://homotopytypetheory.org/book`.

The code can be found here `https://github.com/owl77/PyHottPure`. Clone the repository and start the CLI with ”python -i proof.py” (python version 3 required). This software is the analogue of PyLog for Dependent Type Theory, it is an assistant for deriving judgments (proving theorems) in dependent type theory. Here are some examples from the theorems included in the repository (warning: some reflect an older version).

```
0.  ⊢   CtxEmp
1.  ⊢ U0 : U1  UIntro 0
2.  ⊢ W : U0  AxInt 1
3.  ⊢ X : U0  AxInt 1
4.  ⊢ Bool : U0  AxInt 1
5. x : X ⊢ Bool : U0  Wkg1 3 4
6.  ⊢ (x:X) -> Bool : U0  PiForm 3 5
7. y : W ⊢ (x:X) -> Bool : U0  Wkg1 2 6
8.  ⊢ (y:W) -> (x:X) -> Bool : U0  PiForm 2 7
>>> NewDef(8,"Prop")
Prop := (y:W) -> (x:X) -> Bool : U0
9.  ⊢ Prop : U0  DefInt 8
```

```
0.  ⊢   CtxEmp
1.  ⊢ U0 : U1  UIntro 0
2. A : U0 ⊢   CtxExt 1
3. A : U0 ⊢ U0 : U1  UIntro 2
4. A : U0, B : U0 ⊢   CtxExt 3
5. A : U0, B : U0 ⊢ A : U0  Vble 4
6. A : U0, B : U0, x : A ⊢   CtxExt 5
7. A : U0, B : U0, x : A ⊢ B : U0  Vble 6
8. A : U0, B : U0 ⊢ (x:A) -> B : U0  PiForm 5 7
9. A : U0, B : U0, f : (x:A) -> B ⊢   CtxExt 8
10. A : U0, B : U0, g : (x:A) -> B ⊢ (x:A) -> B : U0  Wkg1 8 8
11. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B ⊢   CtxExt 10
12. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B ⊢ A : U0  Vble 11
13. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B, x : A ⊢   CtxExt 12
14. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B, x : A ⊢ x : A  Vble 13
15. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B, x : A ⊢ f : (x:A) -> B  Vble 13
16. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B, x : A ⊢ f x : B  PiElim 15 14
17. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B, x : A ⊢ g : (x:A) -> B  Vble 13
18. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B, x : A ⊢ g x : B  PiElim 17 14
19. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B, x : A ⊢ B : U0  Vble 13
20. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B, x : A ⊢ equals B f x g x : U0  EqForm 19 16 18
```

```
21. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B ⊢ A : U0  Vble 11
22. A : U0, B : U0, g : (x:A) -> B, f : (x:A) -> B ⊢ (x:A) -> equals B f x g x : U0  PiForm 21 20
23. A : U0, B : U0, g : (x:A) -> B ⊢ (f:(x:A) -> B) -> (x:A) -> equals B f x g x : U0  PiForm 10 22
24. A : U0, B : U0 ⊢ (g:(x:A) -> B) -> (f:(x:A) -> B) -> (x:A) -> equals B f x g x : U0  PiForm 8 23
25. A : U0, B : U0 ⊢ (g:(x:A) -> B) -> (f:(x:A) -> B) -> (x:A) -> equals B f x g x : U1  UCumul 24
26. A : U0 ⊢ (B:U0) -> (g:(x:A) -> B) -> (f:(x:A) -> B) -> (x:A) -> equals B f x g x : U1  PiForm 3 25
27.  ⊢ (A:U0) -> (B:U0) -> (g:(x:A) -> B) -> (f:(x:A) -> B) -> (x:A) -> equals B f x g x : U1  PiForm 1 26

hot := (A:U0) -> (B:U0) -> (g:(x:A) -> B) -> (f:(x:A) -> B) -> (x:A) -> equals B f x g x

28.  ⊢ hot : U1  DefInt 27


0.  ⊢   CtxEmp
1.  ⊢ U0 : U1  UIntro 0
2. A : U0 ⊢   CtxExt 1
3. A : U0 ⊢ A : U0  Vble 2
4. A : U0, a : A ⊢   CtxExt 3
5. A : U0, a : A ⊢ A : U0  Vble 4
6. A : U0, a : A, x : A ⊢   CtxExt 5
7. A : U0, a : A, x : A ⊢ x : A  Vble 6
8. A : U0, a : A ⊢ a : A  Vble 4
9. A : U0, a : A ⊢ λ(x:A)x a ≡ a : A  PiComp 7 8
10. A : U0, x : A ⊢   CtxExt 3
11. A : U0, x : A ⊢ A : U0  Vble 10
12. A : U0 ⊢ (x:A) -> A : U0  PiForm 3 11
13. A : U0, y : (x:A) -> A ⊢   CtxExt 12
14. A : U0, y : (x:A) -> A ⊢ A : U0  Vble 13
15. A : U0, y : (x:A) -> A, a : A ⊢   CtxExt 14
16. A : U0, y : (x:A) -> A, a : A ⊢ y : (x:A) -> A  Vble 15
17. A : U0, y : (x:A) -> A, a : A ⊢ a : A  Vble 15
18. A : U0, y : (x:A) -> A, a : A ⊢ y a : A  PiElim 16 17
19. A : U0, x : A ⊢   CtxExt 3
20. A : U0, x : A ⊢ x : A  Vble 19
21. A : U0 ⊢ λ(x:A)x : (x:A) -> A  PiIntro 20
22.  ⊢ λ(A:U0)λ(x:A)x : (A:U0) -> (x:A) -> A  PiIntro 21

>>> NewDef(22,"id")
id := λ(A:U0)λ(x:A)x : (A:U0) -> (x:A) -> A

23. C : U0, A : U0 ⊢ λ(x:A)x : (x:A) -> A  Wkg1 1 21
24. C : U0 ⊢   CtxExt 1
25. C : U0 ⊢ C : U0  Vble 24
26. C : U0 ⊢ λ(A:U0)λ(x:A)x C ≡ λ(x:C)x : (x:C) -> C  PiComp 23 25
27. C : U0 ⊢   CtxExt 1
28. C : U0 ⊢ C : U0  Vble 27
29. C : U0, x : C ⊢   CtxExt 28
30. C : U0, x : C ⊢ C : U0  Vble 29
31. C : U0 ⊢ (x:C) -> C : U0  PiForm 28 30
32. C : U0, y : (x:C) -> C ⊢   CtxExt 31
33. C : U0, y : (x:C) -> C ⊢ C : U0  Vble 32
34. C : U0, y : (x:C) -> C, a : C ⊢   CtxExt 33
35. C : U0, y : (x:C) -> C, a : C ⊢ y : (x:C) -> C  Vble 34
36. C : U0, y : (x:C) -> C, a : C ⊢ a : C  Vble 34
37. C : U0, y : (x:C) -> C, a : C ⊢ y a : C  PiElim 35 36
38. C : U0, a : C ⊢ λ(A:U0)λ(x:A)x C a ≡ λ(x:C)x a : C  Subst3 26 37
39. C : U0, a : C ⊢   CtxExt 28
40. C : U0, a : C ⊢ C : U0  Vble 39
41. C : U0, a : C, x : C ⊢   CtxExt 40
42. C : U0, a : C, x : C ⊢ x : C  Vble 41
43. C : U0, a : C ⊢ a : C  Vble 39
```

```
44. C : U0, a : C ⊢ λ(x:C)x a ≡ a : C  PiComp 42 43
45. C : U0, a : C ⊢ λ(A:U0)λ(x:A)x C a ≡ a : C  Trans 38 44
46. C : U0, a : C ⊢ id C a ≡ a : C  DefInt 45
```

```
>>> ShowProof()
0. z : Σ(x:A)B ⊢ C z : U0  NewTheorem
1. x : A,y : B ⊢ w : C pair x y  NewTheorem
2.  ⊢ c : Σ(x:A)B  NewTheorem
3.  ⊢ ind Σ(x:A)B (λz.C z) (λx.y.w) c : C c  SigmaElim 0 1 2
4.  ⊢ a : A  NewTheorem
5.  ⊢ b : B  NewTheorem
6.  ⊢ ind Σ(x:A)B (λz.C z) (λx.y.w) pair a b ≡ w : C pair a b  SigmaComp 0 1 4 5
7. x : A ⊢ λ(y:B)w : Π(y:B)C pair x y  PiIntro 1
8.  ⊢ λ(x:A)λ(y:B)w : Π(x:A)Π(y:B)C pair x y  PiIntro 7
9.  ⊢   CtxEmp
10.  ⊢ U3 : U4  UIntro 9
11.  ⊢ U3 : U5  UCumul 10
12.  ⊢ d : C c  NewTheorem
13.  ⊢ pair c d : Σ(z:Σ(x:A)B)C z  SigmaIntro 0 2 12

>>> NewDef("Magma", "Sigma U0 lambda A. (Pi A lambda x. (Pi A lambda x. A  ))")
Magma := Σ(A:U0)Π(x:A)Π(x:A)A

>>> ShowProof()
0. z : plus A B ⊢ C z : U0  NewTheorem
1. x : A ⊢ c x : C inl x  NewTheorem
2. y : B ⊢ d y : C inr y  NewTheorem
3.  ⊢ e : plus A B  NewTheorem
4.  ⊢ ind plus A B (λz.C z) (λx.c x) (λy.d y) e : C e  PlusElim 0 1 2 3
5.  ⊢ a : A  NewTheorem
6.  ⊢ ind plus A B (λz.C z) (λx.c x) (λy.d y) inl a ≡ c a : C inl a  PlusComp 0 1 2 5 1
7.  ⊢ b : B  NewTheorem
8.  ⊢ ind plus A B (λz.C z) (λx.c x) (λy.d y) inr b ≡ d y : C inr b  PlusComp 0 1 2 7 2

>>> ShowProof()
0.  ⊢   CtxEmp
1.  ⊢ 0 : Nat  NatIntro1 0
2.  ⊢ succ 0 : Nat  NatIntro2 1
3.  ⊢ succ succ 0 : Nat  NatIntro2 2
4.  ⊢ succ succ succ 0 : Nat  NatIntro2 3
5. x : Nat ⊢ C x : U0  NewTheorem
6.  ⊢ a : C 0  NewTheorem
7. x : Nat,y : C x ⊢ c x y : C succ x  NewTheorem
8.  ⊢ b : Nat  NewTheorem
9.  ⊢ ind Nat (λx.C x) a (λxy.c x y) b : C b  NatElim 5 6 7 8
10.  ⊢ ind Nat (λx.C x) a (λxy.c x y) succ b ≡ c b ind Nat (λx.C x) a (λxg.c x g) b : C succ b  NatComp2 5 6 7 8
11.  ⊢ ind Nat (λx.C x) a (λxy.c x y) 0 ≡ a : C 0  NatComp2 5 6 7

>>> ShowProof()
0. x : O ⊢ C x : U0  NewTheorem
1.  ⊢ a : O  NewTheorem
2.  ⊢ ind O (λx.C x) a : C a  OElim 0 1

>>> ShowProof()
0. x : A,y : A,p : equals A x y ⊢ C x y p : U0  NewTheorem
1. z : A ⊢ c z : C z z refl z  NewTheorem
2.  ⊢ a : A  NewTheorem
3.  ⊢ b : A  NewTheorem
4.  ⊢ w : equals A a b  NewTheorem
5.  ⊢ ind equals A (λxyp.C x y p) (λz.c z) a b w : C a b w  EqElim 0 1 2 3 4
```

6. $\vdash$ `ind equals A (`λ`xyp.C x y p) (`λ`z.c z) a a refl a` $\equiv$ `c a : C a a refl a  EqComp 0 1 2`

References

[1] Protin, C. L. (2022). Introduction to PyLog. http://dx.doi.org/10.48550/arXiv.2304.02074.