an Intensional Dependent Type Theory with Type-in-Type and Recursion

February 13, 2021

1 Examples

1.1 pretending $\star =_\star \bot$

spoofing an equality

$(\lambda pr : (\star =_\star \bot) . pr\ (\lambda x.x)\ \bot \quad : \neg \star =_\star \bot)\, refl_{\star:\star}$

elaborates to

$(\lambda pr : (\star =_\star \bot) . pr\ (\lambda x.x)\ \bot \quad : \neg \star =_\star \bot)\, (refl_{\star:\star} :: (\star =_\star \star) =_l (\star =_\star \bot))$

$refl_{\star:\star} :: (\star =_\star \star) =_l (\star =_\star \bot)\ (\lambda x.x)\ \bot \quad :\bot$

$(\lambda C : (\star \to \star) . \lambda x : C\ \star . x :: (\Pi C : (\star \to \star) . C\,\star \to C\,\star) =_l (\Pi C : (\star \to \star) . C\,\star \to C\ \bot))\ (\lambda x.x)\ \bot$ $:\bot$

$(\lambda x : \star . x :: (\star \to \star) =_{l,bod} (\star \to \bot))\ \bot \quad :\bot$

$(\bot :: \star =_{l,bod.bod} \bot) \quad :\bot$

note that the program has not yet "gotten stuck". to exercise this error, $\bot$ must be eliminated, this can be done by tying to summon another type by applying it to $\bot$

$((\bot :: \star =_{l,bod.bod} \bot) \quad :\bot)\ \star$

$((\Pi x : \star . x) :: \star =_{l,bod.bod} (\Pi x : \star . x))\ \star$

the computation is stuck, and the original application can be blamed on account that the "proof" has a discoverable type error

$refl_{\star:\star} : \star =_\star \bot$

$\lambda C : (\star \to \star) . \lambda x : C\ \star . x \quad : \Pi C : (\star \to \star) . C\,\star \to \underline{C\ \bot}$

when

$C \coloneqq \lambda x.x$

$C\ \bot = \bot \neq \star = C\,\star$

2 TODO

- syntax and rules
- Example of why function types alone are underwhelming
  - pair,singleton

- walk through of varous examples

- theorem statements
  - substitution!
- proofs
- exposition
- archive
- paper target

3 Scratch

$S_{a:A} \coloneqq a$

$S_{a:A} \coloneqq \mathit{\Pi} P : A \to \star . P\, a \to \star$

..

$\neg \star =_\star (\star \to \star)$ is provable?

$\lambda pr : (\mathit{\Pi} C : (\star \to \star) . C\, Unit \to C\ \bot) . pr\ (\lambda x . x)\ \bot \quad : \neg \star =_\star \bot$

.

evaluating $\neg true_c =_{\mathbb{B}_c} false_c$ with an incorrect proof.

$(\lambda pr : (\mathit{\Pi} C : (\mathbb{B}_c \to \star) . C\, true_c \to C\, false_c) . pr\ (\lambda b : \mathbb{B}_c . b \star Unit\ \bot)\ tt \quad : \neg true_c =_{\mathbb{B}_c} false_c)\, refl_{true_c:\mathbb{B}}$

is elaborated to

$(\lambda pr : (\mathit{\Pi} C : (\mathbb{B}_c \to \star) . C\, true_c \to C\, false_c) . pr\ (\lambda b : \mathbb{B}_c . b \star Unit\ \bot)\ tt \quad : \neg true_c =_{\mathbb{B}_c} false_c)\, (refl_{true_c:}$

$\rightsquigarrow ((refl_{true_c:\mathbb{B}_c} :: true_c =_{\mathbb{B}_c} true_c =_l true_c =_{\mathbb{B}_c} false_c)\ (\lambda b : \mathbb{B}_c . b \star Unit\ \bot)\ tt \quad : \bot)$

de-sugars to

$(((\lambda C : (\mathbb{B}_c \to \star) . \lambda x : C\, true_c . x) :: (\mathit{\Pi} C : (\mathbb{B}_c \to \star) . C\, true_c \to C\, true_c) =_l (\mathit{\Pi} C : (\mathbb{B}_c \to \star) . C\, true_c \to C\, f$

$\rightsquigarrow (((\lambda x : (true_c \star Unit\ \bot) . x) :: (true_c \star Unit\ \bot \to true_c \star Unit\ \bot) =_{l,bod} ((true_c \star Unit\ \bot) \to false_c \star$

def eq to

$(((\lambda x : Unit . x) :: (Unit \to Unit) =_{l,bod} (Unit \to \bot))\ tt \quad : \bot)$

$\rightsquigarrow (tt :: Unit =_{l,bod.bod} \bot \quad : \bot)$

note that the program has not yet "gotten stuck". to exercise this error, $\bot$ must be eliminated, this can be done by tying to summon another type by applying it to $\bot$

$(tt :: Unit =_{l,bod.bod} \bot \quad : \bot)\ \bot \quad : \bot$

bad attempt

$(tt :: Unit =_{l,bod.bod} \bot \quad : \bot)\ \bot \quad : \bot$

de-sugars to

$((\lambda A : \star . \lambda a : A . a) :: \mathit{\Pi} A : \star . A \to A =_{l,bod.bod}\ \mathit{\Pi} x : \star . x \quad : \bot)\ \bot \quad : \bot$

$\rightsquigarrow (.\lambda a . a) :: \bot \to \bot =_{l,bod.bod.bod} \bot \quad : \bot$

but still

$((.\lambda a . a) :: \bot \to \bot =_{l,bod.bod.bod} \bot \quad : \bot) \star$

$((.\lambda a . a) :: \bot \to \bot =_{l,bod.bod.bod}\ \mathit{\Pi} x : \star . x \quad : \bot) \star$

$(\star :: \star =_{l,bod.bod.bod.aty} \bot =_{l,bod.bod.bod.bod} \star)$

bad attempt

$(tt :: Unit =_{l,bod.bod} \perp \quad :\perp) \star \to \star \quad : \star \to \star$
de-sugars to
$((\lambda A : \star.\lambda a : A.a) :: \mathit{\Pi} A : \star.A \to A =_{l,bod.bod} \mathit{\Pi} x : \star.x \quad :\perp) \star \to \star \quad : \star \to \star$
$\rightsquigarrow (.\lambda a.a) :: (\star \to \star) \to (\star \to \star) =_{l,bod.bod.bod} \star \to \star \quad : \star \to \star$
not yet "gotten stuck"
$\rightsquigarrow (.\lambda a.a) :: (\star \to \star) \to (\star \to \star) =_{l,bod.bod.bod} \star \to \star \quad : \star \to \star$
bad attempt
de-sugars to
$((\lambda A : \star.\lambda a : A.a) :: \mathit{\Pi} A : \star.A \to A =_{l,bod.bod} \mathit{\Pi} x : \star.x \quad :\perp) \mathbb{B}_c \quad : \mathbb{B}_c$
$\rightsquigarrow (.\lambda a : \mathbb{B}_c.a) :: \mathbb{B}_c \to \mathbb{B}_c =_{l,bod.bod.bod} \mathbb{B}_c \quad : \mathbb{B}_c$
attempt
$(tt :: Unit =_{l,bod.bod} \perp \quad :\perp) \mathbb{B}_c \quad : \mathbb{B}_c$
de-sugars to
$((\lambda A : \star.\lambda a : A.a) :: \mathit{\Pi} A : \star.A \to A =_{l,bod.bod} \mathit{\Pi} x : \star.x \quad :\perp) \mathbb{B}_c \quad : \mathbb{B}_c$
$\rightsquigarrow (.\lambda a : \mathbb{B}_c.a) :: \mathbb{B}_c \to \mathbb{B}_c =_{l,bod.bod.bod} \mathbb{B}_c \quad : \mathbb{B}_c$
.
.
.
.
.
.