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

February 18, 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 at the point of application l

$\lambda \Pi C : (\star \to \star) . C \star \to \underline{C \star} \neq \lambda \Pi C : (\star \to \star) . C \star \to \underline{C \ \bot}$

when

$C \coloneqq \lambda x.x$

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

1.2 pretending $true_c =_{\mathbb{B}_c} false_c$

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

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

is elaborated to

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

$(refl_{true_c : \mathbb{B}_c} :: true_c =_{\mathbb{B}_c} true_c =_{l} true_c =_{\mathbb{B}_c} false_c) \ (\lambda b : \mathbb{B}_c . b \ \star \ \star \ \bot) \ \bot$

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

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

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

As in the above the example has not yet "gotten stuck". As above, applying $\star$ will discover the error, which would result in an error like

$\lambda \mathit{\Pi} C : (\mathbb{B}_c \to \star)\,.C\,true_c \to \underline{C\,true_c} \neq \lambda \mathit{\Pi} C : (\mathbb{B}_c \to \star)\,.C\,false_c \to \underline{C\,false_c}$

when

$C \coloneqq \lambda x.x$

$C\,true_c = \perp \neq \star = C\,false_c$

2 Proof summery

elaboration produces a well typed term

type soundness goes through as before but, all head constructors match or blame is availiblein an empty ctx

3 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

4 Scratch

4.1 pretending $\lambda x.x =_{\mathbb{B}_c \to \mathbb{B}_c} \lambda x.true_c$

a difference can be observed via

$(\lambda pr : (\mathit{\Pi} C : ((\mathbb{B}_c \to \mathbb{B}_c) \to \star)\,.C\ (\lambda x.x) \to C\ (\lambda x.true_c))\,.pr\ (\lambda f : \mathbb{B}_c \to \mathbb{B}_c.f\ \star\ \star\ \perp)\ \perp \qquad : \neg \lambda x.x =_{\mathbb{B}_c \to \mathbb{B}}$

...

$S_{a:A} \coloneqq a$

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

..

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

$\lambda pr : (\Pi C : (\star \to \star) . C\, Unit \to C\ \ \bot) . pr\ (\lambda x . x)\ \bot \qquad : \neg\star =_\star \bot$

.

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

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

is elaborated to

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

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

de-sugars to

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

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

def eq to

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

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

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 $\bot$

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

bad attempt

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

de-sugars to

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

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

but still

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

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

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

bad attempt

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

de-sugars to

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

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

not yet “gotten stuck”

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

bad attempt

de-sugars to

$((\lambda A : \star . \lambda a : A . a) :: \Pi A : \star . A \to A =_{l,bod.bod}\ \Pi x : \star . x \qquad :\bot)\, \mathbb{B}_c \qquad : \mathbb{B}_c$

$\rightsquigarrow (.\lambda a : \mathbb{B}_c . a) :: \mathbb{B}_c \to \mathbb{B}_c =_{l,bod.bod.bod} \mathbb{B}_c \qquad : \mathbb{B}_c$

attempt

$(tt :: Unit =_{l,bod.bod} \bot \qquad :\bot)\, \mathbb{B}_c \qquad : \mathbb{B}_c$

de-sugars to

$((\lambda A : \star . \lambda a : A . a) :: \Pi A : \star . A \to A =_{l,bod.bod}\ \Pi x : \star . x \qquad :\bot)\, \mathbb{B}_c \qquad : \mathbb{B}_c$

$\rightsquigarrow (.\lambda a : \mathbb{B}_c . a) :: \mathbb{B}_c \to \mathbb{B}_c =_{l,bod.bod.bod} \mathbb{B}_c \qquad : \mathbb{B}_c$

.

.