Encoding Structural Equality in CaPriCon

Marc Coiffier

Contents

```
'utils require import
• Required module: utils

Type 'A -> A 'x ->

'Eq_context { A 'a -> Type ? '.Eq -> .Eq ( x ) '.refl -> } def

'Eq A 'y -> Eq_context .Eq ( y ) ? ? "x = y" defconstr !

'refl Eq_context .refl ! ! "refl x" defconstr

The type of \lambda(y:A) (e: x = y).\mu(e) is \forall(y: A) (e: x = y) (Eq^P: \forall(a: A), x = a \rightarrow Set_1), Eq^P x (refl x) \rightarrow Eq^P ye .

2 lambdas [ 'Eq 'refl ] { export } each
```