# 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 $\lambda(\mathrm{y} : \mathrm{A})\,(\mathrm{e} : \mathrm{x} = \mathrm{y}).\,\mu(\mathrm{e})$ is $\forall(\mathrm{y} : \mathrm{A})\,(\mathrm{e} : \mathrm{x} = \mathrm{y})\,(\mathrm{Eq}^P : \forall(\mathrm{a} : \mathrm{A}),\, \mathrm{x} = \mathrm{a} \to Set_1),\, \mathrm{Eq}^P\,\mathrm{x}\,(refl\,\mathrm{x}) \to \mathrm{Eq}^P\,\mathrm{y}\,\mathrm{e}$ .

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