Лекция 9

Ilya Yaroshevskiy

9 ноября

Содержание

1 Равенство **1**

2 Элиминаторы **1**
2.1 transport 2
2.2 Контруэнтность 2
2.3 Неравенство 2

1 Равенство

Определение. a, b из типа A, тогда $a \overset{A}{=\!=} b$ (равны в типе A), если сущесвует путь $a \rightsquigarrow b$.

- I — интервальный тип [`left`;`right`]
- Сущесвует непрерывная функция $f : I \to A$
  - f `left` $=_\beta a$
  - f `right` $=_\beta b$

```
\data Path
    | path (f: I -> A) : f left = f right
```

2 Элиминаторы

Пример. Элиминатор для дизъюнкции: `case` $(f : L \to \Theta)$ $(g : R \to \Theta)$ $(v : L \vee R)$: Θ

Определение. Элиминатор для равенства (или для интервального типа I):

```
\func coe (P : I -> \Type) (a : P left) (i : I) : P i \elim i
    | left => a
```

Это не настоящее определение, внутри происходит магия

Примечание. Что здесь написано: матчимся только по `left`. Все объекты вдоль пути равны, значит наверное можем всегда выдавать левую часть. Если `a : P left` (a выполнено в $P\ left$) и `a = a'`, тогда `a' : P right`.

2.1 transport

```
\func transport {A : \Type}  (B : A -> \Type) {a a' : A} (p : a=a') (b : B a) : B a'
    => coe (\lam i => B (p @ i)) b right
```

Примечание. Что здесь написано:

- Есть тип в котором мы живем A
- Нечто что нам выдаст утверждение B
- `p: a=a'`
  - `p @ left` — `a`
  - `p @ right` — `a'`
- Лямбда идет в первый аргумент `coe`, получаем путь из `B a` в `B a'`
- У нас есть `B a`, попросим конец пути `B a'`

```
\func inv (A : \Type) {a a' : A} (p : a = a') : a' = a
    => transport (\lam x => x=a) p (idp {A} {a})
```

Примечание. Константный путь, соединяет a с a

```
\func idp {A : \Type} {a : A} : a = a => path (\lam _ => a)
```

2.2 Контруэнтность

```
\func pmap (A B : \Type) (f : A -> B) (a a' : A) {p : a = a'} : f a = f a'
    => transport (\lam x => f a = f x) p idp
```

2.3 Неравенство

$0 \neq 1$

```
\data Nat
    | zero
    | suc (k : Nat)
```

Докажем что `zero` $\neq$ `suc zero`

- `Not (zero = suc zero)`
- `Not (A : \Type) : A -> Empty`

```
\func proof_ne (a : Nat) : \Type \elim a
    | zero => 0 = 0
    | succ x => Empty
\func zne (x : 0 = 1) : Empty => transport proof_ne {0} {1} x idp
```