

&

Дизайн и реализация
языка программирования
с обобщенными множествами,
типами и отображениями
в качестве значений
первого класса

PLC 2017

Квачев В.Д.
ЮФУ


```
int myFunction(const MyType& x);
```

`myFunction :: MyType → IO Int`

`myFunction :: Int → Int → Int`

Refinement Types

- LiquidHaskell

LiquidHaskell

```
{-@ type Between a b = {v:Int | a <= v &&  
                             v < b} @-}  
  
{-@ myValue :: Between 0 10 @-}  
myValue      = 0
```



```
myFunction (x : MyType1) : MyType2 = y
```

Дизайн и реализация
языка программирования
с обобщенными множествами,
типами и отображениями
в качестве значений
первого класса

PLC 2017

Квачев В.Д.
ЮФУ

Типы как множества

Типы как множества

$x : \text{Int}$

$y : \{1, 2, 3, 4, 5\}$

Операции над множествами: пересечение

`x : Int & {1, 2, 3}`

Операции над множествами: объединение

`MyValue = Int | String | [MyValue]`

Предикаты как множества

Предикаты как множества

$c : \text{isLower} = 'a'$

$y < 3$

$y = 0$

Типы \leftrightarrow множества \leftrightarrow предикаты

$$\{ z : \text{Nat} \mid z < 10 \}$$
$$\rightarrow$$
$$\text{Nat} \ \& \ (<10)$$

Типы \leftrightarrow множества \leftrightarrow ???

Обратные функции

`myDecrement (x + 1) = x`

Обратные функции

`myFunction (isLowercase str) = 42`

Предикат?

```
isOdd x = if x % 2 = 1  
          then True  
          else False
```

Предикат?

`isOdd x = if mod x 2 = 1
 then x`

Обратные функции

```
myFunction (x : isOdd) = div (x - 1) 2
```


Обратные функции

```
myFunction (x : isOdd) = div (x - 1) 2  
myFunction (isOdd x)   = (x - 1) :div: 2
```

Обратные функции

`myFunction (isOdd x + 1) = x :div: 2`

Обратные функции

`inverse (++x) = ... // что-то очень сложное`

```
myParser (v ++ " : " ++ isLowercase t)
  : (String, String)
  = (v, t)
```

Конечные множества

`x > 0`

`x < 10`

`xs = {x}`

`print (show xs)`

Градualная типизация

```
a : Nat
x <- read (>a)
x : None    => ()
    Just b => a < b
                print "Success! {a} = "
                print (show {a})
```

Спасибо!

Прототипная реализация typechecker'a
<https://github.com/Rasie1/c-of-x>