

Haskell の Arrow の紹介

初心者 Haskell 勉強会

日比野 啓

2011 年 05 月 25 日

今回の流れ

- Arrowってどんなもの
- Arrow でどんなことができるのか
- Arrow を作るには

今回の流れ

- Arrowってどんなもの
- Arrow でどんなことができるのか
- Arrow を作るには

Arrowってどんなもの

適当に説明すると

Arrow とは入力があって出力があるもの

Arrowってどんな型

例えば関数は Arrow です

```
foo0 :: b -> c
```

```
foo1 :: (->) b c
```

```
-- (->) は引数の2 Type Constructor
```

```
-- Either a b とか見たことがありますよね？
```

```
-- (->) を Arrow a であるような a と置き換えると
```

```
foo2 :: Arrow a => a b c
```

今回の流れ

- Arrowってどんなもの
- Arrow でどんなことができるのか
- Arrow を作るには

Arrow でどんなことができるのか

データフローっぽいものが書けます
論理回路とか

Arrow であるようなオブジェクトが用意されているときは
arr で関数から変換することができます

```
arr :: Arrow a => (b -> c) -> a b c
```


回路の例 - and



- Arrow 記法
- ここで Auto は Arrow になっています

```
and2 :: (Bool, Bool) -> Bool
and2 = uncurry (&&)
```

```
and' :: Auto (Bool, Bool) Bool
and' = proc (i0, i1) -> do
    o0 <- arr and2 -< (i0, i1)
    id -< o0
```

回路の例 - and



- Arrow 記法
- ここで Auto は Arrow になっています

```
and2 :: (Bool, Bool) -> Bool
and2 = uncurry (&&)
```

```
and'' :: Auto (Bool, Bool) Bool
and'' = proc (i0, i1) -> do
    arr and2 -< (i0, i1)
```

回路の例 - and



- Arrow 記法
- ここで Auto は Arrow になっています

```
and2 :: (Bool, Bool) -> Bool
and2 = uncurry (&&)
```

```
and''' :: Auto (Bool, Bool) Bool
and''' = arr and2
```

回路の例 - half adder

```
xor :: Auto (Bool, Bool) Bool
xor = proc i -> do
    m0 <- nand -< i
    m1 <- or'   -< i
    and' -< (m0, m1)
```

```
hAdder :: Auto (Bool, Bool) (Bool, Bool)
hAdder =
    proc i -> do
        o0 <- xor   -< i
        o1 <- and'  -< i
        id  -< (o1, o0)
```

- Arrowってどんなもの
- Arrow でどんなことができるのか
- Arrow を作るには

