

114: myeq x y = (==x) y が正しく動かない

↑ up

- issued: 2020-11-04
- 分類: A サンプルコードが fail
- status: Closed (2020-11-05)

概要

lex が正しい値を返さないことを追っていくと、どうも、elem が従来からちゃんと動いていなかった模様 (常に真を返す)。

elem x = any (== x) なのだが、どうも、関数が x を引数としてとったときの (==x) が正しく動作していないよう。

myeqtest.hs:

```
myeq x y = (==x) y
main = print $ 1 'myeq' 9
```

これが True となる。

調査ログ

2020-11-04 (Wed)

(+x) でも同じようなことが起こるか試してみると、どうも、f x y = (+ x) y が f x y = (+y) y のように化けてしまっているようだ。

以下のプログラム (myaddtest.hs) が 20 になる :

```
myadd x y = (+x) y
main = print $ undefined 'myadd' 10
```

α 変換があやしいかな。core をよく見るとわかりそう。

2020-11-05 (Thu)

Rename において、(== x) が \x -> (==) x x に変換されていたのが原因だった。新しく生成された lambda の仮引数が x だったので名前の衝突を起こしていた。

```

--- a/compiler/src/Rename.hs
+++ b/compiler/src/Rename.hs
@@ -976,12 +976,12 @@ renExp (A.ExpWithTySig e sig) =
    in renExp (A.LetExp [x_tysigdecl, x_valdecl] (A.VarExp x_name))

renExp (A.SectionR opname e) =
- let x_var = A.VarExp (Name "x" (0,0) False)
+ let x_var = A.VarExp (Name "_x#" (0,0) False)
    e_body = A.InfixExp x_var opname e
    in renExp (A.LamExp [x_var] e_body)

renExp (A.SectionL e opname) =
- let x_var = A.VarExp (Name "x" (0,0) False)
+ let x_var = A.VarExp (Name "_x#" (0,0) False)
    e_body = A.InfixExp e opname x_var
    in renExp (A.LamExp [x_var] e_body)

```

他にも同様の問題があるかもしれない。