

## 035: myeq3.hs がランタイムで Error: Non-exhaustive patterns.

↑ up

- issued: 2020-04-16
- 分類: A サンプルコードが fail
- status: Closed (2020-05-15)

### 調査ログ

2020-05-14 (Thu)

もうすこし簡単な例 (casewithwild.hs) でも再現する:

```
myshow (True, True) = "true"
myshow _           = "false"
main = putStrLn $ myshow (False, True)
```

この変換結果 (core0) が、mixture rule によるものになっていない気がする。

```
(Main.myshow :: ((Prelude.(,) Prelude.Bool) Prelude.Bool) -> [Prelude.Char]) =
  \(_Main.myshow.U1 :: ((Prelude.(,) Prelude.Bool) Prelude.Bool)) ->
    case (_Main.myshow.U1 :: ((Prelude.(,) Prelude.Bool) Prelude.Bool)) (_Main.myshow.U1b :: (P
      DEFAULT (_Main.myshow.U1 :: t12)->
        "false"
      Prelude.(,) (_Main.myshow.U2 :: t4) (_Main.myshow.U3 :: t5) :: (t4 -> (t5 -> ((Prelude.(
        case (_Main.myshow.U2 :: t4) (_Main.myshow.U2b :: t4) of
          Prelude.False  :: Prelude.Bool ->
            (Prim.neErr :: t11)
          Prelude.True   :: Prelude.Bool ->
            case (_Main.myshow.U3 :: t5) (_Main.myshow.U3b :: t5) of
              Prelude.False  :: Prelude.Bool ->
                (Prim.neErr :: t10)
              Prelude.True   :: Prelude.Bool ->
                "true"
```

DEFALUT に "false" が入っているあたりが怪しいな。パタンマッチコンパイル周辺で、変数一つの場合を、なんか、特別扱いしてなかったかな。(遠い記憶)

2020-05-15 (Fri)

TrCore における扱い

どこかで Alts をひっくり返している奴がいるなどと思ったら…、TrCore.dsgAlts 内で、rmWild しているところかな…と思ったが、最後に reverse している。どこだろう？

Semant.renProg で逆転していたので、以下のように trace を仕掛けたところ、toBg で逆転していた (tbs は ok, bgs で逆転)

```
let bgs = toBg $ tbs ++ itbs
    bgs'' = toBg $ ctbs ++ tbs ++ itbs
trace ("tbs:" ++ show tbs) $ return ()
trace ("bgs:" ++ show bgs) $ return ()
```

toBg で逆転しているのを直すと、それにもなって TrCore.renExpr もなおさないと、一次的に動かなくなると思われる。ブランチ切ろう。

手順としては、toBg での逆転をなおす、rmWild でわざと逆転させる、で、ひとまずいままで通り動く。そ  
のご、Fatbar を変換をただしくなおす、という感じかな。

逆転していた箇所： BindGrouping.bindMap

```
--- a/compiler/src/BindGrouping.hs
+++ b/compiler/src/BindGrouping.hs
@@ -127,7 +127,7 @@ bindMap tbs' rh = bmap tbs' rh Map.empty
     i = fromMaybe (error $ "Must not happen. Vertex Id not found: " ++ name)
         (Map.lookup name h)
     tb' = case Map.lookup i d of
-       Just (_, _, alts') -> (name, Nothing, alts++alts')
+       Just (_, _, alts') -> (name, Nothing, alts'++alts)
       Nothing             -> tb

     d' = Map.insert i tb' d
```

自分を前に足していた。

解決

TrCore.transExpr も修正して、ただしく case 式が生成されるようになった。

ただし、すでに通っていた sample194 (showlist2.hs) が通らなくなってしまったので、これは再び testases

に降格する。また、この `showlist2.hs` にあるのと同様の定義が、`instance Show [Double—Float]` に残っていたのが同じくエラーしていたが、これらは不要だったので `lib/Prelude.hs` から削除。

なお、本件の修正においては、`transExpr` に `Fatbar e1@(case ..) e2` というのが来たら、`e1` の `DEFALUT` に `e2` を設定するのかなと思っていたのだが、そもそもこの形で `transExpr` に来ることがなかった。このあたり、要復習。