

097: eqcls.hs で signature too general

↑ up

- issued: 2020-05-30
- 分類: A サンプルコードが fail
- status: Closed (2020-05-30)

概要

lib/Prelude.hs にて Maybe を定義しようとしてでくわしたエラーの再現。testcases/eqcls.hs で signature too general となる。

eqcls.hs:

```
data Hoge a = Fuga | Hoge a

meq :: (Eq t) => (Hoge t) -> (Hoge t) -> Bool
Fuga  'meq' Fuga  = True
Hoge x 'meq' Hoge y = x == y
_      'meq' _      = False

main = print $ Fuga 'meq' Hoge 5
```

ただ、lib/Prelude.hs では、instance (Show a) => Show [a] など、同程度に複雑な型シグネチャはすでに通っているのだが、list, tuple など特殊ケースのみだったので、一般的なクラス定義の変換経路になんらかの不備があるのかもしれない。

eqcls.hs と似たリスト版では通るので、その変換過程と比較して調査するのがいいかと思う。

eqcls2.hs:

```
meq :: (Eq t) => [t] -> [t] -> Bool
[]  'meq' [] = True
[x] 'meq' [y] = x == y
_   'meq' _   = False

main = do print $ [3] 'meq' [2]
          print $ [5] 'meq' [5]
```

調査ログ

2020-05-30 (Sat)

-ddump-ren オプションで、Rename 結果をダンプさせるようにしてみた。meq の型シグネチャの部分だけ抜き出してみたが、両方で同じ構造の型シグネチャが生成されているように見える。

```
Forall [Star] ([IsIn "Prelude.Eq" (TGen 0)] :=> TAp (TAp (TCon (Tycon "(->)" (Kfun Star (Kfun Star S
```

```
Forall [Star] ([IsIn "Prelude.Eq" (TGen 0)] :=> TAp (TAp (TCon (Tycon "(->)" (Kfun Star (Kfun Star S
```

解決

型シグネチャに問題ないなら、環境の方に問題があることになる。Prelude.[] については PreDefined.hs で型コンストラクタ、値コンストラクタともに定義しているのだが、Hoge は Rename がコンストラクタを生成しているので、そのあたりが怪しい。

Rename で値コンストラクタの型スキームを生成している箇所で、型変数を量化していなかったのがよくなかったようだ：

```
diff --git a/compiler/src/Rename.hs b/compiler/src/Rename.hs
index 872b166..9af2d61 100644
--- a/compiler/src/Rename.hs
+++ b/compiler/src/Rename.hs
@@ -168,7 +168,7 @@ scanDecls ds = do
     let renCs (n, ts) = do
           qn <- renameVar n
           let t' = foldr fn t ts
-         return $ qn >: toScheme t'
+         return $ qn >: (quantify (tv t') ([>: t']))

     as <- mapM renCs cs
     appendCMs (fromAssumpList as)
```

これで通るようになった。lib/Prelude.hs における Maybe の定義もおおかた OK (Functor のみなんかエラーしている)。