

118: (Monad m) => Int -> m Int の Kind を推定できない

↑ up

- issued: 2020-12-11
- 分類: A サンプルコードが fail
- status: Open

概要

以下のようなプログラムをコンパイルしようとする、Kind の推定に失敗してエラーする:

```
$ cat returnint.hs
retInt :: (Monad m) => Int -> m Int
retInt = return

main = do
  x <- retInt 10
  print x
```

2021-10-27 (Wed)

全件確認中:

```
unno@unno-FMVD70GN7G ~/work/bissues/118
$ cat returnint.hs
retInt :: (Monad m) => Int -> m Int
retInt = return

main = do
  x <- retInt 10
  print x
unno@unno-FMVD70GN7G ~/work/bissues/118
$ runhaskell returnint.hs
10
unno@unno-FMVD70GN7G ~/work/bissues/118
$ ~/prj/bunny/compiler/bin/bunny testrun returnint.hs
```

```

/home/unno/prj/bunny/compiler/bin/bunnyc -d ./jout/returnint --xno-implicit-prelude /home/unno/prj/b
/home/unno/prj/bunny/compiler/bin/bunnyc -d ./jout/returnint --xlibrary-path /home/unno/prj/bunny/co
warning: kiexpr' :AppTy (Tyvar (Name {origName = "m", namePos = (1,31), isConName = False})) (Tycon
bunnyc: Kind not infered ("m",[])
CallStack (from HasCallStack):
  error, called at src/Rename.hs:728:14 in main:Rename
CallStack (from -prof):
  Rename.kindLookup (src/Rename.hs:(727,1)-(728,77))
  Rename.renSigvar.k (src/Rename.hs:678:7-32)
  Rename.renSigvar (src/Rename.hs:(674,1)-(688,51))
  Rename.renDecls.renDecl (src/Rename.hs:(631,5)-(645,50))
  Rename.renDecls (src/Rename.hs:(583,1)-(645,50))
  Semant.renProg (src/Semant.hs:(21,1)-(52,11))
  Main.doCompile(...) (app/Main.hs:83:7-70)
  Main.doCompile (app/Main.hs:(77,1)-(114,33))
  Main.main (app/Main.hs:(117,1)-(132,42))
testrun: failed to compile returnint.hs

```

2022-09-18 (Sun)

現象の確認。

```

unno@Noether 118 % pwd
/Users/unno/work/bissues/118
unno@Noether 118 % cat returnint.hs
retInt :: (Monad m) => Int -> m Int
retInt = return

main = do
  x <- retInt 10
  print x

unno@Noether 118 % ~/prj/bunny/compiler/bin/bunny testrun returnint.hs
/Users/unno/prj/bunny/compiler/bin/bunnyc -d ./jout/returnint --xno-implicit-prelude /Users/unno/prj
/Users/unno/prj/bunny/compiler/bin/bunnyc -d ./jout/returnint --xlibrary-path /Users/unno/prj/bunny/
warning: kiexpr' :AppTy (Tyvar (Name {origName = "m", namePos = (1,31), isConName = False})) (Tycon
bunnyc: Kind not infered ("m",[])
CallStack (from HasCallStack):

```

```
error, called at src/Rename.hs:731:14 in main:Rename
testrun: failed to compile returnint.hs
```

当然ながら再現する。returnint を少し改変して、型注釈を省くと通る。

```
unno@Noether 118 % cat returnint2.hs
retInt = return
```

```
main = do
  x <- retInt 10
  print x
```

```
unno@Noether 118 % ~/prj/bunny/compiler/bin/bunny testrun returnint2.hs
/Users/unno/prj/bunny/compiler/bin/bunnyc -d ./jout/returnint2 --xno-implicit-prelude /Users/unno/pr
/Users/unno/prj/bunny/compiler/bin/bunnyc -d ./jout/returnint2 --xlibrary-path /Users/unno/prj/bunny
10
```

この種推論については、

- (\rightarrow) の各引数の種は $*$ である
- Int の種は $*$ である

ことから、 $\text{Int} \rightarrow m \text{Int}$ の部分のみから m の種は推定できるべきだろうから、(種に関する調査をしな
ければ、というのではなく) 修正方針は立てられそうなものだけど、すっかり遠い過去に書いたコードになっ
てしまっているので、当該ソースを読んできれいにするところから取り掛からないと。

” Kind not infered” と言っているのは、src/Rename.hs