

## 112: triple2.hs で qname not found

↑ up

- issued: 2020-10-11
- 分類: A サンプルコードが fail
- status: Closed (2020-12-29)

### 概要

triple2.hs (以下) をコンパイルすると、`bunncyc: qname not found: "a"` となる。

```
main =  
  let (a, b, c) = (1, 2, 3)  
  in print a
```

なお、`print a` を `print "foo"` などにかえて、`a` を参照しないようにすると、今度はランタイムエラーとなる。

memo:

- 3つ組じゃなく、pair にしたらどうかなど試してみよう

### 調査ログ

2020-11-06 (Fri)

pair なら問題なし:

```
main =  
  let (a, b) = (1, 2)  
  in print (a+b)
```

2020-12-28 (Mon)

タプルに限らず、3つ以上の引数をとるパターンがだめみたい。たとえば、つぎのような:

```
data Triple a b c = Triple a b c
```

```
main = let Triple a b c = Triple 1 2 3
      in print a
```

Pair a b だと OK だった。これで被疑箇所はかなり絞れると思う。これを対処したときには、同様のパターンで複数の引数をとるパターンをざーっと試験しよう。こんなの（これを16引数くらいまではやろう）:

```
data X a b c d = X a b c d
main = let X a b c d = X 1 2 3 4
      in print d
```

2020-12-29 (Tue)

犯人は、ここ (Rename.hs の一部):

```
trAsPat n (A.FunAppExp (A.FunAppExp c a) b) rhs = do
  let d1 = [A.ValDecl (A.VarExp n) rhs]
      a1 = A.VarExp (Name "_a#1" (0,0) False)
      a2 = A.VarExp (Name "_a#2" (0,0) False)
      cab = A.FunAppExp (A.FunAppExp c a1) a2
      e2 = A.CaseExp (A.VarExp n) [A.Match cab (A.UnguardedRhs a1 [])]
      e3 = A.CaseExp (A.VarExp n) [A.Match cab (A.UnguardedRhs a2 [])]
      d2 = case a of
        A.WildcardPat -> []
        _               -> [A.ValDecl a (A.UnguardedRhs e2 [])]
      d3 = case b of
        A.WildcardPat -> []
        _               -> [A.ValDecl b (A.UnguardedRhs e3 [])]
  in return $ concat [d1, d2, d3]
```

この場合、いつも最後の2変数のみ束縛できちゃうんだな。実際、つぎのようなプログラムは現状でも動作する:

```
main =
  let (a, b, c) = (1, 2, 3)
  in print c
```

ちなみに、`let [a, b, c] = [1, 2, 3]` みたいなケースは、`let (a : ( b : ( c : []))) = [1, 2, 3]` なので、2引数でカバーされていた (`sample203.hs`)。

以下のように修正：

```
-      trAsPat n (A.FunAppExp (A.FunAppExp c a) b) rhs = do
-        let d1 = [A.ValDecl (A.VarExp n) rhs]
-            a1 = A.VarExp (Name "_a#1" (0,0) False)
-            a2 = A.VarExp (Name "_a#2" (0,0) False)
-            cab = A.FunAppExp (A.FunAppExp c a1) a2
-            e2 = A.CaseExp (A.VarExp n) [A.Match cab (A.UnguardedRhs a1 [])]
-            e3 = A.CaseExp (A.VarExp n) [A.Match cab (A.UnguardedRhs a2 [])]
-            d2 = case a of
-              A.WildcardPat -> []
-              _             -> [A.ValDecl a (A.UnguardedRhs e2 [])]
-            d3 = case b of
-              A.WildcardPat -> []
-              _             -> [A.ValDecl b (A.UnguardedRhs e3 [])]
-        in return $ concat [d1, d2, d3]
+      trAsPat n e@(A.FunAppExp _ _) rhs = do
+        let parsef (A.FunAppExp f@(A.FunAppExp _ _) a) vs = parsef f (a:vs)
+            parsef (A.FunAppExp c a) vs = (c, a:vs)
+            (c, vs) = parsef e []
+            d0 = A.ValDecl (A.VarExp n) rhs
+            as = [A.VarExp (Name ("_a#" ++ show i) (0,0) False)
+                  | i <- [1..(length vs)]]
+            cas = foldl' A.FunAppExp c as
+            ds = [case v of
+                  A.WildcardPat -> []
+                  _ -> [A.ValDecl v (A.UnguardedRhs e [])]
+                  | (v, a) <- zip vs as
+                  , let e = A.CaseExp (A.VarExp n) [A.Match cas (A.UnguardedRhs a [])]
+                  ]
+        in return $ (d0:concat ds)
```

`sample324, 325, 326, 327` を追加。