

088: Binding Groups について調べる

↑ up

- issued: 2020-05-22
- 分類: 分類: D 要調査
- status: Open

概要

Binding Groups について調べる。

現状の toBg 実装は、Thih の仕様をみたしていないように見える。

11.6.3 結合されたバインディング・グループ*1 より:

「es 内の束縛は、グループ内のどの束縛に依存してもよいが、少なくとも isn に含まれるものには依存するはずである。そうでないなら、グループはおそらく最小になっていない。また、es が空なら n=1 になる」

073 の問題は、これにかかわっているようにも思えるが、まだわからない。

以下の記事なども参照すべきか:

- 16.1.1.5. Typechecking of recursive binding groups*2
- Haskell 2010 4.5.1 Dependency Analysis*3

2020-05-22 (Fri)

073 の対策のなかで、tcBind の Type-check の結果 subst を引き継ぐように修正:

```
if null qs then ((v, e'), st{tcNum=num, tcSubst=s})
else ((v, Lam (mkVs n qs) e'), st{tcNum=num, tcSubst=s})
```

さらに、概要で引用した BindGroup の要件を鑑み、Expl が null のときには、[[Impl]] のサイズが 1 になるように、let binding をばらす仮対処をいれた (Rename):

```
@@ -703,7 +705,12 @@ renExp (A.LetExp ds e) = do
  e' <- renExp e
  exitLevel
  let bgs = toBg tbs
```

*1 <https://uhideyuki.sakura.ne.jp/studs/index.cgi/ja/thihnote#p26>

*2 https://downloads.haskell.org/~ghc/latest/docs/html/users_guide/bugs.html#typechecking-of-recursive-binding-groups

*3 <https://www.haskell.org/onlinereport/haskell2010/haskellch4.html#x10-890004.5.1>

```

- assert (length bgs == 1) $ return (Let (head bgs) e') -- TODO: (head bgs) is temporary
+   (es, iss) = head bgs
+   f []      bdy = bdy
+   f (is:iss') bdy = Let ([],[is]) (f iss' bdy)
+ if null es
+   then return $ f iss e'
+   else return (Let (head bgs) e')

```

これによって、068 でいれていた仮対処、つまり、AsPat の処理で、以下のように case の検査項に直接右辺値を書いていた回避策を解除できた。

```

x = (1, 2)
a = case (1, 2) of
  (x, y) -> x

```