077: Num を Eq, Show ２つのサブクラスにする

↑ up

- issued: 2020-05-15
- 分類： B 機能追加
- status: Closed (2020-05-16)

概要

現状では、２つ以上のスーパークラスに対応していないため、Num は Eq のサブクラスとなっており、Show のサブクラスになっていない。

なので、以下のようなプログラムにおいて、f, g の型は以下のようになってしまっている：

- Main.f :: Forall [Star] ([Prelude.Num a] :=> (a -> (a -> ((Prelude.(,) a) Prelude.Bool))))
- Main.g :: Forall [Star] ([Prelude.Num a,Prelude.Show a] :=> (a -> (a -> (Prelude.IO ()))))

```
f x y = let z = x * y
            c = x == y
        in (z, c)

g x y = print (x * y)

main = do print $ f 1 2
          print $ f (1::Double) 1
          g 3 3
          g (3::Double) 10
```

f は正しく (Num a) => a -> a -> (a, Bool) だが、g の方が (Num a, Show a) => a -> a -> IO () となってしまっている。

これは、Num が Show のサブクラスでもあるなら、(Num a) => a -> a -> IO () となる。

調査ログ

2020-05-16 (Sat)

ソースの読み取り以降の部分は、複数のスーパクラスに対応するように作ってあったので、微修正だけで OK だった。

```
index 938764d..961e04c 100644
--- a/compiler/lib/Prelude.hs
+++ b/compiler/lib/Prelude.hs
@@ -192,7 +192,7 @@ instance Show Char where
 -- todo: should convert to printable characters
 showLitChar c = (++) [c]

-class (Eq a) => Num a where
+class (Eq a, Show a) => Num a where
   (+), (-), (*) :: a -> a -> a
   negate        :: a -> a
   signum        :: a -> a
diff --git a/compiler/src/Rename.hs b/compiler/src/Rename.hs
index 8e78cfb..d45e99e 100644
--- a/compiler/src/Rename.hs
+++ b/compiler/src/Rename.hs
@@ -245,6 +245,8 @@ renClassDecls dcls = do
   where
     extr_sc Nothing                                  = []
     extr_sc (Just (A.ParTy (A.AppTy (A.Tycon i) _))) = [origName i]
+    extr_sc (Just (A.TupleTy ts))                    =
+      map (\(A.AppTy (A.Tycon i) _) -> origName i) ts
     extr_sc (Just t) = error $ "extr_sc: " ++ show t

     clsadd (maybe_sc, A.AppTy (A.Tycon n) _) = do
```

修正後の f, g の型 :

- Main.f :: Forall [Star] ([Prelude.Num a] :=> (a -> (a -> ((Prelude.(,) a) Prelude.Bool))))
- Main.g :: Forall [Star] ([Prelude.Num a] :=> (a -> (a -> (Prelude.IO ()))))