Is there a type error in the Haskell code:

v = let f x = x in (f 3, f '3')

Yes, there is a type error in the given Haskell code snippet:

```
v = let f x = x in (f 3, f '3')
```

The problem arises with the usage of f in the tuple (f 3, f '3'). Here, f is first applied to 3, which is an integer, hence f is inferred to have the type f :: Int → Int for this usage. Immediately after, f is applied to '3', which is a character. This implies that f should also have the type f :: Char → Char. A function in Haskell cannot have more than one type unless explicitly using advanced features like Type Families or other extensions which are not in use here by default.