

This error occurs when the compiler is unable to unambiguously infer the return type of a function or method which is generic on return type, such as the `collect` method for `Iterator`s.

For example:

```
fn main() {  
    let n: u32 = 1;  
    let mut d: u64 = 2;  
    d = d + n.into();  
}
```

Here we have an addition of `d` and `n.into()`. Hence, `n.into()` can return any type `T` where `u64: Add<T>`. On the other hand, the `into` method can return any type where `u32: Into<T>`.

The author of this code probably wants `into()` to return a `u64`, but the compiler can't be sure that there isn't another type `T` where both `u32: Into<T>` and `u64: Add<T>`.

To resolve this error, use a concrete type for the intermediate expression:

```
fn main() {  
    let n: u32 = 1;  
    let mut d: u64 = 2;  
    let m: u64 = n.into();  
    d = d + m;  
}
```