A trait implementation has stricter requirements than the trait definition.

Erroneous code example:

```
trait Foo {
    fn foo<T>(x: T);
}

impl Foo for bool {
    fn foo<T>(x: T) where T: Copy {}
}
```

Here, all types implementing `Foo` must have a method `foo<T>(x: T)` which can take any type `T`. However, in the `impl` for `bool`, we have added an extra bound that `T` is `Copy`, which isn't compatible with the original trait.

Consider removing the bound from the method or adding the bound to the original method definition in the trait.