

A lifetime bound on a trait implementation was captured at an incorrect place.

Erroneous code example:

```
trait Id<T> {}
trait Lt<'a> {}

impl<'a> Lt<'a> for () {}
impl<T> Id<T> for T {}

fn free_fn_capture_hrtb_in_impl_trait()
    -> Box<for<'a> Id<impl Lt<'a>>> // error!
{
    Box::new(())
}

struct Foo;
impl Foo {
    fn impl_fn_capture_hrtb_in_impl_trait()
        -> Box<for<'a> Id<impl Lt<'a>>> // error!
    {
        Box::new(())
    }
}
```

Here, you have used the inappropriate lifetime in the `impl Trait`, The `impl Trait` can only capture lifetimes bound at the `fn` or `impl` level.

To fix this we have to define the lifetime at the function or `impl` level and use that lifetime in the `impl Trait`. For example you can define the lifetime at the function:

```
trait Id<T> {}
trait Lt<'a> {}

impl<'a> Lt<'a> for () {}
impl<T> Id<T> for T {}

fn free_fn_capture_hrtb_in_impl_trait<'b>()
    -> Box<for<'a> Id<impl Lt<'b>>> // ok!
{
    Box::new(())
}

struct Foo;
impl Foo {
    fn impl_fn_capture_hrtb_in_impl_trait<'b>()
        -> Box<for<'a> Id<impl Lt<'b>>> // ok!
```

```
    {  
      Box::new()  
    }  
}
```