

An `async` function used recursion without boxing.

Erroneous code example:

```
async fn foo(n: usize) {
    if n > 0 {
        foo(n - 1).await;
    }
}
```

To perform async recursion, the `async fn` needs to be desugared such that the `Future` is explicit in the return type:

```
use std::future::Future;
fn foo_desugared(n: usize) -> impl Future<Output = ()> {
    async move {
        if n > 0 {
            foo_desugared(n - 1).await;
        }
    }
}
```

Finally, the future is wrapped in a pinned box:

```
use std::future::Future;
use std::pin::Pin;
fn foo_recursive(n: usize) -> Pin<Box<dyn Future<Output = ()>>> {
    Box::pin(async move {
        if n > 0 {
            foo_recursive(n - 1).await;
        }
    })
}
```

The `Box<...>` ensures that the result is of known size, and the pin is required to keep it in the same place in memory.