The type does not fulfill the required lifetime.

Erroneous code example:

```
use std::sync::Mutex;

struct MyString<'a> {
    data: &'a str,
}

fn i_want_static_closure<F>(a: F)
    where F: Fn() + 'static {}

fn print_string<'a>(s: Mutex<MyString<'a>>) {

    i_want_static_closure(move || {     // error: this closure has lifetime 'a
                                        //        rather than 'static
        println!("{}", s.lock().unwrap().data);
    });
}
```

In this example, the closure does not satisfy the `'static` lifetime constraint. To fix this error, you need to double check the lifetime of the type. Here, we can fix this problem by giving `s` a static lifetime:

```
use std::sync::Mutex;

struct MyString<'a> {
    data: &'a str,
}

fn i_want_static_closure<F>(a: F)
    where F: Fn() + 'static {}

fn print_string(s: Mutex<MyString<'static>>) {

    i_want_static_closure(move || {     // ok!
        println!("{}", s.lock().unwrap().data);
    });
}
```