Rust Quiz 解读： Quiz 11

Quiz 11:

下面这段代码输出什么?

```
fn f<'a>() {}
fn g<'a: 'a>() {}

fn main() {
    let pf = f::<'static> as fn();
    let pg = g::<'static> as fn();
    print!("{}", pf == pg);
}
```

输出结果： 编译错误

解读

考察要点:

1. 生命周期参数概念
2. 生命周期参数限定：`Early bound` VS `Late bound`
3. 生命周期子类型与协变
4. 函数指针及其比较

不得不说，此Quiz代码中涉及一个隐晦的概念：生命周期参数`Early bound` VS `Late bound`。这两个概念是官方提供的书里没有过的，也是我做这个题首次遇到的概念。通过调查Rust源码，大概得出了它们的定义:

```
// 'a , 'c 是Late bound，而'c 是Early bound
fn foo<'a,'b,'c,T:Trait<'b>>(...)

// 'b 是Late Bound，而'c 是Early bound
fn bar<'b, 'c>(x: &self, y: &'b u32, z: &'c u64) where 'static: 'c

// 结构体S和枚举体E中的'a 和 'b 就是Early bound
struct S<'a, 'b>(&'a u8, &'b u8);
enum E<'a, 'b> {
    V(&'a u8),
    U(&'b u8),
}
```

还有很多例子，就不举例了。可以总结出来：

`Early Bound`，是指用于限定类型（或其他生命周期参数）的生命周期参数。比如：`'long : 'short`，这其中`'short`就是`Early bound`。

`Late Bound`，只要不是`Early bound`，那么剩下的就是`Late bound`。

回到我们的Quiz代码中。看得出来，`fn f<'a>() {}`中`'a`就是`Late bound`生命周期参数。而`fn g<'a: 'a>() {}`中`'a`就是`Early bound`生命周期参数。

那么区分这两种类型的生命周期参数有什么意义呢？我总结了以下几点：

1. 如果没有`Late bound`参数，那么编译器基本可以推断类型的生命周期为`'static`的。

2. 如果是`Early bound`参数，那意味着允许子类型协变。也就是说，`'static`可以替换`'a`。因为`'static`是`'a`的子类型。和第一条结论一致。
3. `Late bound`参数，意味着「不变（invariant）」，也就是说，不协变，也不逆变。

所以，在main函数中，`let pf = f::<'static> as fn();`，这样编译器报错：`error: cannot specify lifetime arguments explicitly if late bo`。也就是说，不能通过turbofish操作符`::<>`指定`f`的生命周期为`'static`，因为它是late bound，它是invariant的。所以，不能为其指定子类型`'static`。将这行代码改成`let pf = f as fn();`就可以正常执行Quiz代码了。

因为`fn g<'a: 'a>() {}`中`'a`是`Early bound`，所以允许协变， main函数中`let pg = g::<'static> as fn();`指定了子类型生命周期参数`'static`并不影响其行为。

但是，最终的比较结果会返回false。这是为什么呢？因为在Rust中，不能随便比较函数指针。即便函数签名是一样的，但出于底层的优化策略，这种比较会产生未定义行为。可以参考这个issues中的比较结果：issues 54685，该issues中比较两个函数指针x和y，在Debug和Release下编译，行为并不一致。

最后，看一下结构体和枚举体中的生命周期参数是不是`Early bound`呢？代码如下：

```
struct S<'a, 'b>(&'a u8, &'b u8);
enum E<'a, 'b> {
    V(&'a u8),
    U(&'b u8),
}

fn main() {
    S(&0, &0); // OK
    S::<'static, 'static>(&0, &0);

    E::V(&0); // OK
    E::V::<'static, 'static>(&0);
}
```

上面代码正常编译运行。即验证了结构体和枚举体中生命周期参数`'a`和`'b`是`Early bound`生命周期参数，允许协变。

点此查看 Rust Quiz 11