An attempt was made to implement `Drop` on a specialization of a generic type.

Erroneous code example:

```
trait Foo {}

struct MyStruct<T> {
    t: T
}

impl<T: Foo> Drop for MyStruct<T> {
    fn drop(&mut self) {}
}
```

This code is not legal: it is not possible to specialize `Drop` to a subset of implementations of a generic type. In order for this code to work, `MyStruct` must also require that `T` implements `Foo`. Alternatively, another option is to wrap the generic type in another that specializes appropriately:

```
trait Foo{}

struct MyStruct<T> {
    t: T
}

struct MyStructWrapper<T: Foo> {
    t: MyStruct<T>
}

impl <T: Foo> Drop for MyStructWrapper<T> {
    fn drop(&mut self) {}
}
```