

Cannot use the associated type of a trait with uninferred generic parameters.

Erroneous code example:

```
pub trait Foo<T> {  
    type A;  
  
    fn get(&self, t: T) -> Self::A;  
}  
  
fn foo2<I : for<'x> Foo<&'x isize>>(  
    field: I::A) {} // error!
```

In this example, we have to instantiate `'x`, and we don't know what lifetime to instantiate it with. To fix this, spell out the precise lifetimes involved. Example:

```
pub trait Foo<T> {  
    type A;  
  
    fn get(&self, t: T) -> Self::A;  
}  
  
fn foo3<I : for<'x> Foo<&'x isize>>(  
    x: <I as Foo<&isize>>::A) {} // ok!  
  
fn foo4<'a, I : for<'x> Foo<&'x isize>>(  
    x: <I as Foo<&'a isize>>::A) {} // ok!
```