An associated type binding was done outside of the type parameter declaration and `where` clause.

Erroneous code example:

```
pub trait Foo {
    type A;
    fn boo(&self) -> <Self as Foo>::A;
}

struct Bar;

impl Foo for isize {
    type A = usize;
    fn boo(&self) -> usize { 42 }
}

fn baz<I>(x: &<I as Foo<A=Bar>>::A) {}
// error: associated type bindings are not allowed here
```

To solve this error, please move the type bindings in the type parameter declaration:

```
# struct Bar;
# trait Foo { type A; }
fn baz<I: Foo<A=Bar>>(x: &<I as Foo>::A) {} // ok!
```

Or in the `where` clause:

```
# struct Bar;
# trait Foo { type A; }
fn baz<I>(x: &<I as Foo>::A) where I: Foo<A=Bar> {}
```