An attempted implementation of a trait method has the wrong number of type or const parameters.

Erroneous code example:

```
trait Foo {
    fn foo<T: Default>(x: T) -> Self;
}

struct Bar;

// error: method `foo` has 0 type parameters but its trait declaration has 1
// type parameter
impl Foo for Bar {
    fn foo(x: bool) -> Self { Bar }
}
```

For example, the `Foo` trait has a method `foo` with a type parameter `T`, but the implementation of `foo` for the type `Bar` is missing this parameter. To fix this error, they must have the same type parameters:

```
trait Foo {
    fn foo<T: Default>(x: T) -> Self;
}

struct Bar;

impl Foo for Bar {
    fn foo<T: Default>(x: T) -> Self { // ok!
        Bar
    }
}
```