

An attempted implementation of a trait method has the wrong number of function parameters.

Erroneous code example:

```
trait Foo {
    fn foo(&self, x: u8) -> bool;
}

struct Bar;

// error: method `foo` has 1 parameter but the declaration in trait `Foo::foo`
// has 2
impl Foo for Bar {
    fn foo(&self) -> bool { true }
}
```

For example, the `Foo` trait has a method `foo` with two function parameters (`&self` and `u8`), but the implementation of `foo` for the type `Bar` omits the `u8` parameter. To fix this error, they must have the same parameters:

```
trait Foo {
    fn foo(&self, x: u8) -> bool;
}

struct Bar;

impl Foo for Bar {
    fn foo(&self, x: u8) -> bool { // ok!
        true
    }
}
```