A negative implementation was marked as unsafe.

Erroneous code example:

```
struct Foo;

unsafe impl !Clone for Foo { } // error!
```

A negative implementation is one that excludes a type from implementing a particular trait. Not being able to use a trait is always a safe operation, so negative implementations are always safe and never need to be marked as unsafe.

This will compile:

```
#![feature(auto_traits)]

struct Foo;

auto trait Enterprise {}

impl !Enterprise for Foo { }
```

Please note that negative impls are only allowed for auto traits.