A lint check attribute was overruled by a `forbid` directive set as an attribute on an enclosing scope, or on the command line with the `-F` option.

Example of erroneous code:

```
#![forbid(non_snake_case)]

#[allow(non_snake_case)]
fn main() {
    let MyNumber = 2; // error: allow(non_snake_case) overruled by outer
                      //        forbid(non_snake_case)
}
```

The `forbid` lint setting, like `deny`, turns the corresponding compiler warning into a hard error. Unlike `deny`, `forbid` prevents itself from being overridden by inner attributes.

If you're sure you want to override the lint check, you can change `forbid` to `deny` (or use `-D` instead of `-F` if the `forbid` setting was given as a command-line option) to allow the inner lint check attribute:

```
#![deny(non_snake_case)]

#[allow(non_snake_case)]
fn main() {
    let MyNumber = 2; // ok!
}
```

Otherwise, edit the code to pass the lint check, and remove the overruled attribute:

```
#![forbid(non_snake_case)]

fn main() {
    let my_number = 2;
}
```