WG14 N2663

Title: **Lifetime, Blocks, and Labels**
Author: **Martin Uecker, University Medical Center Göttingen**
Date: **2021-02-13**

N2508 introduced changes to allow placing of labels before declarations and at the end of compound statements. These changes were accepted into C23. Unfortunately, there is an unintended side effect which affects the storage duration of some objects in rare cases.

Consider the following example:

```
int x = 1;
int *p = &x;
a:
 if (*p && (p = &(int){ 0 }))
   goto a;
```

This example appears to be well-defined because lifetime of the compound literal does not end when control is transferred back to the `if` statement from the `goto` statement. With the change introduced in N2508 this is translated as if the label were followed by a null statement:

```
int x = 1;
int *p = &x;
a: ;
 if (*p && (p = &(int){ 0 }))
   goto a;
```

This change makes no difference in terms of control flow, but has an unintended side effect: The control flow after the `goto` then leaves the block of the `if` statement that defines the life time of the compound statement. This causes the pointer `p` to become indeterminate and the following access then becomes undefined behavior. The following proposed wording change prevents this.

Suggested Wording Changes

The green color denotes the change already introduced N2508 into the latest draft (N2596) and the new proposed wording is underlined.

6.8.2

Semantics

A compound statement is a block. **A label that is not followed by a statement inside the same compound statement shall be translated as if it were followed by a null statement.**

Acknowledgment: twitter user void friend (@rep_stosq_void) for bringing this up.