C static code analysis: The three expressions of a "for" statement should only be concerned with loop control

3 minutes

`for` loops are very flexible in C and C++. Because of that, they can carry a complexity that can make the code error-prone, difficult to understand, and hard to maintain.

Many for loops can be written in a way that clearly separates the iteration process from the content of the iteration. This rule makes sure that all the code relevant to the iteration is placed in the for-loop header. The compliant code is then easier to reason about.

A for loop is composed of 4 sub-parts:

```
for([initialization]; [condition]; [update])
  [body]
```

We classify the variables used to control them in three categories:

- A *loop-counter* is a variable modified in the update. It should not be modified in the body.
- A *loop-constant* is an auxiliary variable declared in the initialization. It's very often used to precompute some data about the end condition or the stride.
- A *pseudo-counter* shares some properties with a loop counter, but its update conditions are more complex. It will therefore only be updated in the body, and cannot be used in the update. Using a pseudo-counter makes the loop more complex to reason about, and therefore is not permitted. They are very often declared in the initialization, for instance, to limit their scope, but in some cases reuse existing variables.

Additionally, the loop condition should refer to at least one *loop-counter*, and should not modify anything.

This rule is only checking for loops with a condition and an update.

Noncompliant Code Example

```
for( int h = 0, int i = 0 ; h < 10 ; i += 1 ) { // Noncompliant, the loop-counter is not used in the condition
}

for( int i = 0 ; i++ < 10 ; i += 1 ) { // Noncompliant, loop-counter i is updated in the condition
}

for( int i = 0 , int h = 0; i+(++h) < 10 ; i += 1 ) { // Noncompliant, pseudo-counter h is updated in the condition
}

for( int i = 0 ; i < 10 ; i += 1 ) { // Noncompliant, loop-counter i is updated in the body
  if (i%2)  { ++i;}
}

for( int i = 0 , j = 0 ; i < 10 ; i += j) { // Noncompliant, pseudo-counter j is is used in the update
  j = i + 1;
}

for( int i = 0 , j = 0 ; i < 10 + j ; i += 1) { // Noncompliant, pseudo-counter j is is used in the condition
  j = i + 1;
}
```

See

- MISRA C:2004, 13.5 - The three expressions of a for statement shall be concerned only with loop control.
- MISRA C++:2008, 6-5-5 - A *loop-control-variable* other than the *loop-counter* shall not be modified within *condition* or *expression*.

- MISRA C:2012, 14.2 - A for loop shall be well-formed