N3371 - Integer and arithmetic constant expressions

Author: Javier A. Múgica

Versions

Initial version: 2024 - oct - 03

Contents

The problem

The standard uses the sentence *the result is an integer constant expression*, applied to **`sizeof`** and **`alignof`** expressions. This is not right, since what is an integer constant expression is the expression itself, as the term conveys; the result in those cases is an integer constant; it is a property of the value of the expression, or of the evaluation of the expression, and exists only if the expression is evaluated. *constant expression* is somthing wich, among other requirements, has the form of a conditional expression, so it cannot be referred to as the result of an evaluation.

“Integer constant” was the term used until recently in those places, which was changed by the adoption of the proposal N3239 *Some constants are literally literals*. As that proposal argues, the use of “integer constant” there was incorrect because this term is defined as a certain category of tokens. That paper changed the term to “integer literal” and changed the sentence *the result is an integer constant* to append *expression* after it. In other places, the proposal changed this to “constants of integer type”.

We like the choice of N3239 of talking about **`sizeof`** and **`alignof`** expressions as integer constant expressions, but has to be expressed in some other form.

Wording

First change

In 6.5.4.5 The **`sizeof`** and **`alignof`** operators, pp. 2 and 3, change

the result is an integer constant expression *to* the expression is an integer constant expression

In 6.6 Constant expressions, pp. 8 and 10 and in appendix J.2, items 50 and 52, change

whose results are integer constant expressions *to* which are integer constant expressions

In 6.9 External definitions, in

— part of the operand of a **`sizeof`** operator whose result is an integer constant expression;

— part of the operand of an **`alignof`** operator whose result is an integer constant expression;

change those items to

— part of the operand of a **`sizeof`** expression which is an integer constant expression;

— part of the operand of an **`alignof`** operator;

Second change

If the first change is accepted, the paragraph on integer constant expressions can be simplified to replace "**`sizeof`** expressions which are integer constants expressions and **`alignof`** expressions" by "integer constant expressions". In the paragraph on arithmetic constant expressions the operands "integer literals" and "character literals" can also be included in the replaced text. With the two changes combined, these two paragraphs would be

8 An *integer constant expression* shall have integer type and shall only have operands that are integer literals, named and compound literal constants of integer type, character literals, integer constant expressions, and floating, named, or compound literal constants of arithmetic type that are the immediate operands of casts. Cast operators in an integer constant expression shall only convert arithmetic types to integer types, except as part of an operand to the typeof operators, **`sizeof`** operator, or **`alignof`** operator.

10 An *arithmetic constant expression* shall have arithmetic type and shall only have operands that are floating literals, named or compound literal constants of arithmetic type and integer constant expressions. Cast operators in an arithmetic constant expression shall only convert arithmetic types to arithmetic types, except as part of an operand to the typeof operators, **`sizeof`** operator, or **`alignof`** operator.

Third change

Finally, we understand that the current wording does not allow an expression like `3` or `'a'` to be an ICE by itself, for example in `int a[3];` or `case 'a':`, since there `3` and `'a'` are not operands of anything. The opposite interpretation; namely, that becuase they are not operands none of those *shall*'s apply to them, would make an ICE of any identifier with integer type, which is obviously not. Even if it is understood that it does allow those expressions, we believe it is clearer to state that literals of the appropriate type are integer/arithmetic constant expressions. We propose the following wording in accordance to this:

8 The following are *integer constant expressions*: integer literals, named and compound literal constants of integer type and character literals. Other integer constant expressions are defined elsewhere in this document.[1]

9 An integer constant expression shall have integer type and shall only have operands that are integer constant expressions and floating, named, or compound literal constants of arithmetic type that are the immediate operands of casts. Cast operators in an integer constant expression shall only convert arithmetic types to integer types, except as part of an operand to the typeof operators, **`sizeof`** operator, or **`alignof`** operator.

11 The following are *arithmetic constant expressions*: integer constant expressions, floating literals and named or compound literal constants of arithmetic type.

12 An *arithmetic constant expression* shall have arithmetic type and shall only have operands that are arithmetic constant expressions. Cast operators in an arithmetic constant expression shall only convert arithmetic types to arithmetic types, except as part of an operand to the typeof operators, **`sizeof`** operator, or **`alignof`** operator.

[1] **`alignof`** expressions and some **`sizeof`** expressions (6.5.4.5).