
Rule 0.17 *⟨Write Values Directly Instead of Calculating⟩*

<pre> [...] contract A { [...] function f(pds) { [...] var = <i>expr</i>; stmts } [...] } </pre>	=	<pre> [...] contract A' { [...] function f(pds) { [...] var = <i>literal</i>; stmts } [...] } </pre>
--	---	--

where

- var* is a variable being assigned a value;
- expr* is a compile-time constant arithmetic or logical expression;
- literal* is the pre-computed literal value equivalent to *expr*;
- pds* are the parameter declarations of function *f*;
- stmts* represents the sequence of statements following the assignment.

provided

- The expression *expr* consists only of compile-time constants and operators;
- The value of *expr* can be computed at compile time;
- literal* is the exact result of evaluating *expr*;
- The expressions *expr* and *literal* are semantically equivalent;
- No overflow or precision loss occurs when pre-computing the value;
- The gas cost of computing *expr* at runtime exceeds the cost of using *literal* directly.

Invariant:

- Let s_i and s'_i be the initial state of *A* and *A'*, respectively.
- Let s_f and s'_f be the state reached by *A* and *A'*, respectively, after *A.f()* and *A'.f()* are executed from s_i and s'_i , respectively.
- Then, the coupling invariant is

$$\forall s_i, s'_i . (s_i = s'_i) \rightarrow (s_f = s'_f)$$
