
Rule 0.13 *(Use Constant Variables for Unchanging Values)*

$[...]$ contract A { $T\ var = value;$ $[...]$ function $f(pds)$ view { $[...]$ $\dots\ var\ ...$ $[...]$ $}$ $[...]$ $}$	$=$	$[...]$ contract A' { $T\ \mathbf{constant}\ var = value;$ $[...]$ function $f(pds)$ pure { $[...]$ $\dots\ var\ ...$ $[...]$ $}$ $[...]$
--	-----	---

where

var is a state variable of type T in contract A ;

$value$ is a compile-time constant expression that initializes var ;

f is a function that reads var ;

pds are the parameter declarations of function f .

provided

The variable var is never modified after initialization;

The value $value$ is known at compile time and does not depend on runtime state;

All functions reading var can be changed from **view** to **pure** if they only access constant variables;

The type T supports constant declaration in Solidity;

No write operations to var exist anywhere in the contract.

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

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