
Rule 0.14 *⟨Use Immutable Variables for Constructor-Set Values⟩*

<pre> [...] contract A { T var; [...] constructor(pds) { var = value; [...] } function f() view { [...] ... var ... } [...] } </pre>	=	<pre> [...] contract A' { T immutable var; [...] constructor(pds) { var = value; [...] } function f() view { [...] ... var ... } [...] } </pre>
---	---	---

where

- var* is a state variable of type *T* in contract *A*;
- value* is an expression evaluated during contract construction;
- pds* are the constructor parameter declarations;
- f* is a function that reads *var*.

provided

- The variable *var* is assigned exactly once in the constructor;
- The variable *var* is never modified after construction;
- The type *T* supports immutable declaration in Solidity;
- No write operations to *var* exist outside the constructor;
- The value assigned to *var* can be determined at deployment time.

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

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