
Rule 0.6 *⟨State Variable Packing⟩*

<pre> [...] contract A { T₁ var₁; T₂ var₂; ... T_n var_n; [...] } </pre>	=	<pre> [...] contract A' { T_{π(1)} var_{π(1)}; T_{π(2)} var_{π(2)}; ... T_{π(n)} var_{π(n)}; [...] } </pre>
--	---	---

where

var_i are state variables of the contract with types T_i for $i = 1, \dots, n$;

π is a permutation of $\{1, 2, \dots, n\}$ that reorders the state variables;

T_i represents types of varying sizes (e.g., **uint256**, **uint128**, **uint8**, **bool**, **address**).

provided

The permutation π optimally packs state variables to minimize the number of storage slots used;

Variables with combined size ≤ 32 bytes are grouped together in the reordered declaration;

The reordering does not affect the logical semantics or initialization order of the contract;

All state variable accesses throughout the contract remain semantically equivalent after reordering.

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

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