
Rule 0.5 *⟨Struct Packing⟩*

<pre> [...] contract A { [...] struct S { T₁ field₁; T₂ field₂; ... T_n field_n; } [...] }</pre>	=	<pre> [...] contract A' { [...] struct S { T_{π(1)} field_{π(1)}; T_{π(2)} field_{π(2)}; ... T_{π(n)} field_{π(n)}; } [...] }</pre>
--	---	---

where

S is a struct type defined in the contract;

$field_i$ are the fields of the struct with types T_i for $i = 1, \dots, n$;

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

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

provided

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

Fields with combined size ≤ 32 bytes are grouped together in the reordered struct;

The reordering does not affect the logical semantics of the contract;

All field accesses in 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

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