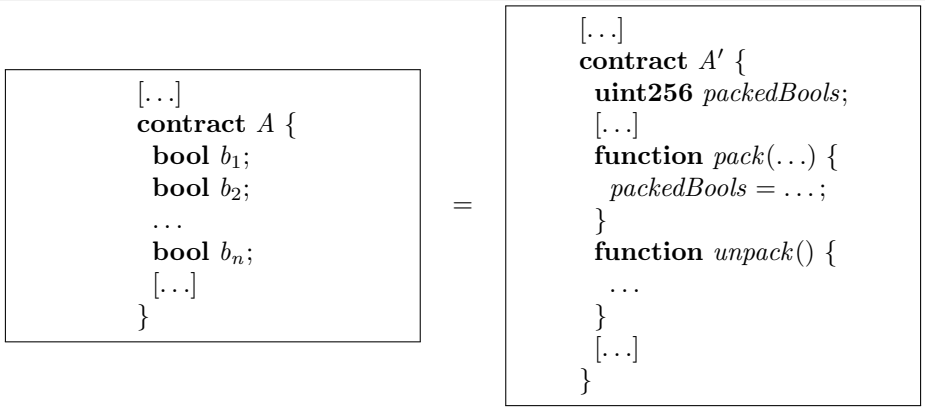
Rule 0.7 ⟨*Boolean Packing*⟩



where

b_i are boolean state variables for $i = 1, \ldots, n$ where $n \leq 32$;

packedBools is a **uint256** variable that stores up to 32 boolean values;

$mask_i = 1 \ll (i-1)$ is the bit mask for the i-th boolean;

pack is a function that sets boolean values in *packedBools* using bitwise operations;

unpack is a function that retrieves boolean values from *packedBools* using bitwise operations.

provided

The number of boolean variables $n \leq 32$;

All accesses to b_i in A are replaced with appropriate calls to *pack* and *unpack* in A';

The packing operation uses: $packedBools = (packedBools \ \& \ \sim mask_i) \ \mid \ (b_i \ ? \ mask_i : 0)$;

The unpacking operation uses: $b_i = (packedBools \ \& \ mask_i) \neq 0$;

No concurrent modifications occur that would violate atomicity of pack/unpack operations.

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

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