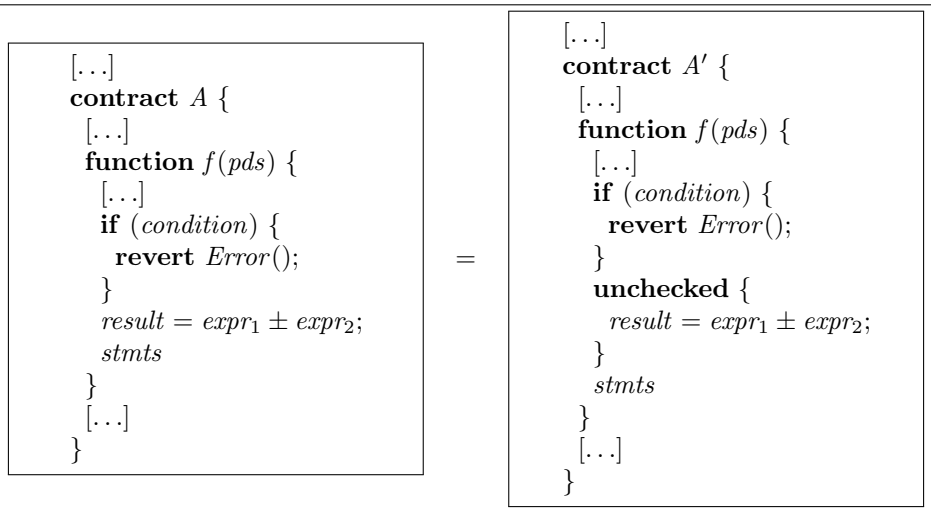
Rule 0.29 ⟨*Use Unchecked Arithmetic for Validated Operations*⟩



where

result is a variable storing the result of an arithmetic operation;

$expr_1$ and $expr_2$ are expressions involved in the arithmetic operation;

$\pm$ represents arithmetic operators ($+$, $-$, $*$, $/$, $\%$, etc.);

condition is a validation that prevents overflow/underflow;

stmts represents the sequence of statements following the arithmetic operation;

pds are the parameter declarations of function f.

provided

Prior validation ensures that overflow/underflow cannot occur;

The condition check guarantees safe arithmetic bounds before the operation;

The arithmetic operation is immediately preceded by validation logic;

Solidity version $\geq 0.8.0$ is used (where overflow checks are enabled by default);

The unchecked block only contains arithmetic operations that are provably safe;

No external calls or state changes occur between validation and arithmetic operation.

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

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