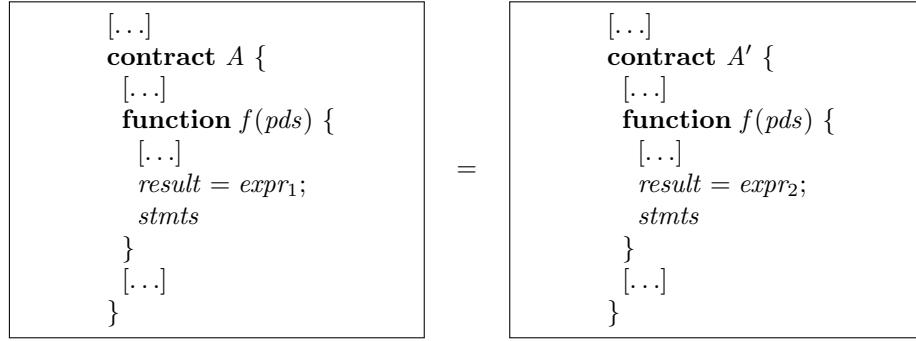

Rule 0.15 *(Reduce Mathematical Expressions)*



where

*expr*₁ is a complex mathematical or logical expression in contract *A*;

*expr*₂ is the reduced form of *expr*₁ with fewer operations;

result is a variable storing the expression result;

pds are the parameter declarations of function *f*;

stmts represents the sequence of statements following the expression.

provided

The expressions *expr*₁ and *expr*₂ are semantically equivalent (evaluate to the same value);

*expr*₂ requires fewer operations than *expr*₁;

The reduction may apply algebraic rules such as: factoring (e.g., $a \cdot x + b \cdot x = (a + b) \cdot x$), De Morgan's laws (e.g., $\neg x \wedge \neg y \equiv \neg(x \vee y)$), distributive properties, or constant folding;

No side effects are introduced or removed by the transformation;

The reduction maintains numerical precision and overflow behavior.

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

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