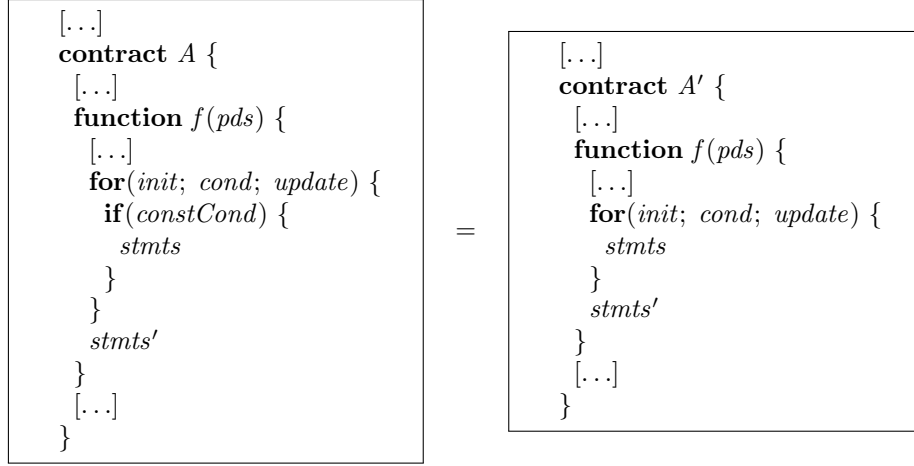


---

**Rule 0.3** *(Refactoring Loops with Constant Comparison)*


---



**where**

*constCond* is a boolean condition that always evaluates to *true* within the loop;  
*init*, *cond*, and *update* are the loop initialization, condition, and update expressions;  
*stmts* represents statements inside the conditional block;  
*stmts'* represents statements following the loop.

**provided**

*constCond* is provably constant and evaluates to *true* for all loop iterations;  
The loop variables and context do not affect the truth value of *constCond*;  
No side effects in *constCond* evaluation;  
*stmts* does not contain control flow statements that would alter loop execution (e.g., **break**, **continue**).

**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

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


---