
Rule 0.30 *⟨Redundant Control Flow Removal (Continue)⟩*

<pre> [...] contract A { [...] function f(pds) { [...] for(init; cond; update) { if(guardCond) continue; stmts } stmts' } [...] } </pre>	=	<pre> [...] contract A' { [...] function f(pds) { [...] for(init; cond; update) { if(¬guardCond) stmts } stmts' } [...] } } </pre>
---	---	--

where

guardCond is a boolean condition that guards the execution of *stmts*;

stmts represents the statements in the loop body that should be conditionally executed;

init, *cond*, and *update* are the loop initialization, condition, and update expressions;

stmts' represents statements following the loop;

$\neg\text{guardCond}$ denotes the logical negation of *guardCond*.

provided

The **continue** statement is immediately executed when *guardCond* is true;

No statements exist after *stmts* and before the end of the loop iteration;

guardCond has no side effects;

stmts does not modify variables used in *guardCond* in a way that would affect the loop's semantics.

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

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