**Rule 0.18** ⟨*Use Single-Line Variable Swapping*⟩

```
[...]
contract $A$ {
  [...]
  function $f(pds)$ {
    [...]
    $T\ temp = var_1$;
    $var_1 = var_2$;
    $var_2 = temp$;
    $stmts$
  }
  [...]
}
```

$=$

```
[...]
contract $A'$ {
  [...]
  function $f(pds)$ {
    [...]
    $(var_1, var_2) = (var_2, var_1)$;
    $stmts$
  }
  [...]
}
```

**where**

$var_1$ and $var_2$ are variables of type $T$ being swapped;

$temp$ is a temporary variable of type $T$ used for the traditional swap;

$T$ is the type of the variables being swapped;

$pds$ are the parameter declarations of function $f$;

$stmts$ represents the sequence of statements following the swap.

**provided**

The variables $var_1$ and $var_2$ are of the same type $T$;

The traditional three-step swap pattern is used in contract $A$;

The temporary variable $temp$ is used only for the swap operation;

No side effects occur during the evaluation of $var_1$ and $var_2$;

The tuple assignment maintains the same semantics as the traditional swap.

**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

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