

# Reasoning Table for operationX

```
void operationX (ContainerOfT& p, T& y);  
  /// updates p  
  /// clears y  
  /// requires  $|p| > 1$   
  /// ensures  $p = \#p * \langle \#y \rangle$ 
```

operationX's *requires* clause goes in *A0*  
operationX's *ensures* clause goes in *CN*

S	Code	Assume	Confirm
0		<i>A0</i>	<i>C0</i>
	<code>p.op1(y);</code>	/ / / / / / / /	/ / / / / / / /
1		<i>A1</i>	<i>C1</i>
	...	/ / / / / / / /	/ / / / / / / /
<i>k</i>		<i>Ak</i>	<i>Ck</i>
	...	/ / / / / / / /	/ / / / / / / /
<i>N</i>		<i>AN</i>	<i>CN</i>

## Reference:

```
void op1 (T& x);  
  /// updates self  
  /// clears x  
  /// requires: self != <>  
  /// ensures:  self = #self * <x>
```

# Reasoning Table for operationX

```
void operationX (ContainerOfT& p, T& y);  
  //! updates p  
  //! clears y  
  //! requires  |p| > 1  
  //! ensures   p = #p * <#y>
```

The *requires* clause for a called operation always appears in the Confirm column in the state previous to the call. For example, *opl*'s requires goes in State 0's Confirm column.

S	Code	Assume	Confirm
0		AO	C0
	p.op1(y);	/ / / / / / / /	/ / / / / / / /
1		A1	C1
	...	/ / / / / / / /	/ / / / / / / /
k		Ak	Ck
	...	/ / / / / / / /	/ / / / / / / /
N		AN	CN

## Reference:

```
void op1 (T& x);  
  //! updates self  
  //! clears x  
  //! requires: self != <>  
  //! ensures:  self = #self * <#x>
```

## Reasoning Table for operationX

```

void operationX (ContainerOfT& p, T& y);
    /// updates p
    /// clears x
    /// requires   |p| > 1
    /// ensures   p = #p * <#y>

```

The *ensures* clause for a called operation always appears in the Assume column in the state subsequent to the call. For example, *opl*'s ensures goes in State 1's Assume column.

S	Code	Assume	Confirm
0		AO	C0
	p.op1(y);	/ / / / / / / /	/ / / / / / / /
1		A1	C1
	..	/ / / / / / / /	/ / / / / / / /
k		Ak	Ck
	...	/ / / / / / / /	/ / / / / / / /
N		AN	CN

### Reference:

```

void op1 (T& x);
    /// updates self
    /// clears x
    /// requires: self != <>
    /// ensures: self = #self * <#x>

```