

# Verification Condition Generation

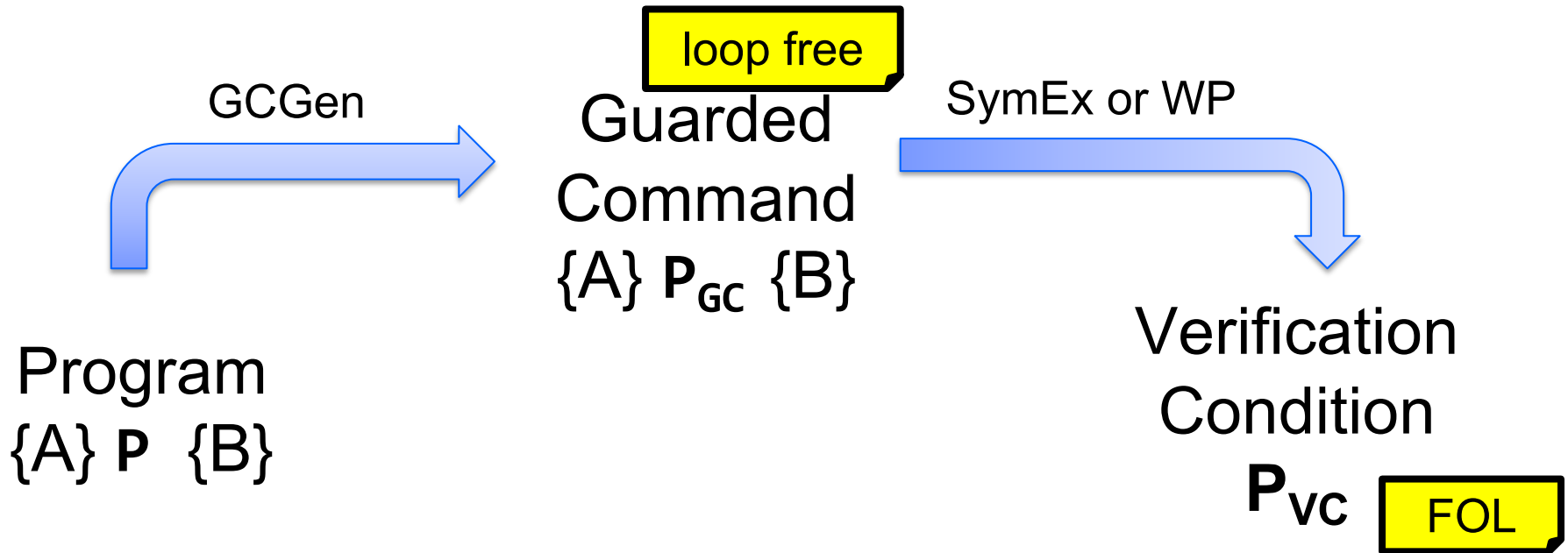
Testing, Quality Assurance, and Maintenance  
Winter 2020

Prof. Arie Gurfinkel

based on slides by Prof. Ruzica Piskac and others



# Verification Condition Generation in a Nutshell



$P_{VC}$  is valid if and only if  $\vdash \{A\} P \{B\}$

# Loop-Free Guarded Commands

Introduce loop-free guarded commands as an intermediate representation of the verification condition

$c ::=$

- | assume  $b$
- | assert  $b$
- | havoc  $x$
- |  $c_1 ; c_2$
- |  $c_1 \parallel c_2$       (*non-deterministic choice*)

# From Programs to Guarded Commands

$GC(\text{skip}) =$

assume true

$GC(x := e) =$

assume  $tmp = x$ ; havoc  $x$ ; assume  $(x = e[tmp/x])$

$GC(c_1 ; c_2) =$

$GC(c_1) ; GC(c_2)$

where  $tmp$  is fresh

$GC(\text{if } b \text{ then } c_1 \text{ else } c_2) =$

$(\text{assume } b; GC(c_1)) \parallel (\text{assume } \neg b; GC(c_2))$

$GC(\text{while } b \text{ inv } I \text{ do } c) = ?$

# Guarded Commands for Loops

```
GC(while  $b$  inv  $I$  do  $c$ ) =  
  assert  $I$ ;  
  havoc  $x_1$ ; ...; havoc  $x_n$ ;  
  assume  $I$ ;  
  ((assume  $b$ ; GC( $c$ ); assert  $I$ ; assume false) ||  
  assume  $\neg b$ )
```

where  $x_1, \dots, x_n$  are the variables modified in  $c$

# Example: VC Generation

$\{n \geq 0\}$

$p := 0;$

$x := 0;$

while  $x < n$  inv  $p = x * m \wedge x \leq n$  do

$x := x + 1;$

$p := p + m$

$\{p = n * m\}$

# Example: VC Generation

Computing the guarded command

$\{ n \geq 0 \}$

assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;

assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;

assert  $p = x * m \wedge x \leq n$ ;

havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ;

( (assume  $x < n$ ;

  assume  $x_1 = x$ ; havoc  $x$ ; assume  $x = x_1 + 1$ ;

  assume  $p_1 = p$ ; havoc  $p$ ; assume  $p = p_1 + m$ ;

  assert  $p = x * m \wedge x \leq n$ ; assume false)

|| assume  $x \geq n$ )

$\{ p = n * m \}$

# Verification Condition Generation

Idea 1: Exhaustive symbolic execution of GC program

- the program is correct if no assertion is ever falsified
- Verification Condition is constructed implicitly by symbolic exec
- Guided by pre-condition and program structure, but not guided by post-condition

Idea 2: propagate the post-condition backwards through the program:

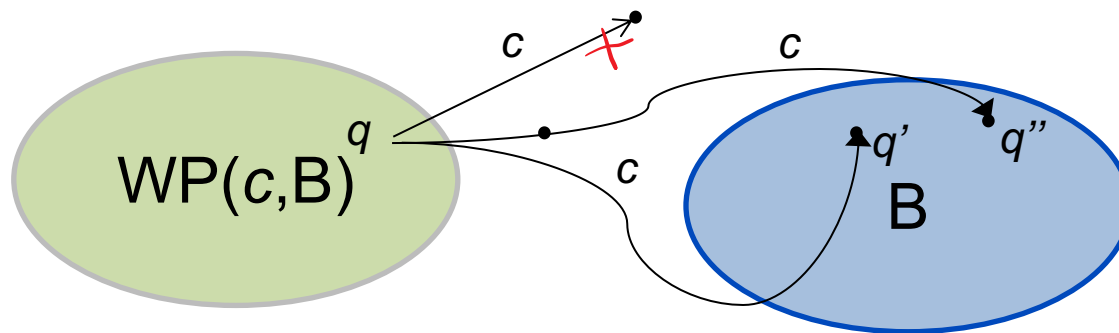
- From a Hoare triple  $\{A\} P \{B\}$
- generate FOL formula  $A \Rightarrow F(P, B)$
- Backwards propagation  $F(P, B)$  is formalized in terms of **weakest preconditions**.



# Weakest Preconditions

The weakest precondition  $WP(c,B)$  holds for any state  $q$  whose  $c$ -successor states all satisfy  $B$ :

$$q \models WP(c,B) \text{ iff } \forall q' \in Q. q \xrightarrow{c} q' \Rightarrow q' \models B$$



Compute  $WP(P,B)$  recursively based on the structure of the program  $P$ .

# Exercises: Weakest Pre-Condition

{ } x := z + w { x >= y }

{ }  
if y > 0 then x := x + 1 else x := y + 4  
{x >= 42}

{ }  
if y > 0 then x := z else x := y  
{x >= y + z}

# Exercises: Weakest Pre-Condition

$\{ z + w \geq y \} x := z + w \{ x \geq y \}$

$\{ \quad \quad \quad \}$   
if  $y > 0$  then  $x := x + 1$  else  $x := y + 4$   
 $\{x \geq 42\}$

$\{ \quad \quad \quad \}$   
if  $y > 0$  then  $x := z$  else  $x := y$   
 $\{x \geq y + z\}$

# Exercises: Weakest Pre-Condition

$\{ z + w \geq y \} x := z + w \{ x \geq y \}$

$\{(y > 0 \wedge x \geq 41) \vee (y \leq 0 \wedge y \geq 38)\}$   
if  $y > 0$  then  $x := x + 1$  else  $x := y + 4$   
 $\{x \geq 42\}$

$\{ \quad \quad \quad \}$   
if  $y > 0$  then  $x := z$  else  $x := y$   
 $\{x \geq y + z\}$

# Exercises: Weakest Pre-Condition

$$\{ z + w \geq y \} x := z + w \{ x \geq y \}$$
$$\{(y > 0 \wedge x \geq 41) \vee (y \leq 0 \wedge y \geq 38)\}$$

if  $y > 0$  then  $x := x + 1$  else  $x := y + 4$   
 $\{x \geq 42\}$

$$\{(y > 0 \wedge z \geq y + z) \vee (y \leq 0 \wedge y \geq y + z)\}$$

if  $y > 0$  then  $x := z$  else  $x := y$   
 $\{x \geq y + z\}$

# Exercises: Weakest Pre-Condition

$$\{ z + w \geq y \} x := z + w \{ x \geq y \}$$
$$\{(y > 0 \wedge x \geq 41) \vee (y \leq 0 \wedge y \geq 38)\}$$

if  $y > 0$  then  $x := x + 1$  else  $x := y + 4$   
 $\{x \geq 42\}$

$$\{(y > 0 \wedge 0 \geq y) \vee (y \leq 0 \wedge 0 \geq z)\}$$

if  $y > 0$  then  $x := z$  else  $x := y$   
 $\{x \geq y + z\}$

# Computing Weakest Preconditions

$$\text{WP}(\text{assume } b, B) = b \Rightarrow B$$

$$\text{WP}(\text{assert } b, B) = b \wedge B$$

$$\text{WP}(\text{havoc } x, B) = B[a/x] \quad (a \text{ fresh in } B) \text{ “replace } x \text{ by } a\text{”}$$

$$\text{WP}(c_1; c_2, B) = \text{WP}(c_1, \text{WP}(c_2, B))$$

$$\text{WP}(c_1 \parallel c_2, B) = \text{WP}(c_1, B) \wedge \text{WP}(c_2, B)$$

# Putting Everything Together

Given a Hoare triple  $H \equiv \{A\} P \{B\}$

Compute  $c_H = \text{assume } A; \text{GC}(P); \text{assert } B$

Compute  $VC_H = \text{WP}(c_H, \text{true})$

Prove  $\vdash VC_H$  using a theorem prover.



# Example: VC Generation

Computing the weakest precondition

```
WP (  assume  $n \geq 0$ ;  
      assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
      assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
      assert  $p = x * m \wedge x \leq n$ ;  
      havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ;  
      ((assume  $x < n$ ;  
        assume  $x_1 = x$ ; havoc  $x$ ; assume  $x = x_1 + 1$ ;  
        assume  $p_1 = p$ ; havoc  $p$ ; assume  $p = p_1 + m$ ;  
        assert  $p = x * m \wedge x \leq n$ ; assume false)  
      || assume  $x \geq n$ ) ;  
      assert  $p = n * m$ , true)
```

# Example: VC Generation

Computing the weakest precondition

WP ( **assume**  $n \geq 0$ ;  
assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
assert  $p = x * m \wedge x \leq n$ ;  
havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ;  
(assume  $x < n$ ;  
assume  $x_1 = x$ ; havoc  $x$ ; assume  $x = x_1 + 1$ ;  
assume  $p_1 = p$ ; havoc  $p$ ; assume  $p = p_1 + m$ ;  
assert  $p = x * m \wedge x \leq n$ ; assume false)  
|| **assume**  $x \geq n$ ,  **$p = n * m$** )

# Example: VC Generation

Computing the weakest precondition

```
WP (  assume  $n \geq 0$ ;  
      assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
      assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
      assert  $p = x * m \wedge x \leq n$ ;  
      havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,  
      WP ((assume  $x < n$ ;  
          assume  $x_1 = x$ ; havoc  $x$ ; assume  $x = x_1 + 1$ ;  
          assume  $p_1 = p$ ; havoc  $p$ ; assume  $p = p_1 + m$ ;  
          assert  $p = x * m \wedge x \leq n$ ; assume false)  
      || assume  $x \geq n$ ,  $p = n * m$ ))
```

# Example: VC Generation

Computing the weakest precondition

WP ( assume  $n \geq 0$ ;  
assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
assert  $p = x * m \wedge x \leq n$ ;  
havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,  
WP (assume  $x < n$ ;  
assume  $x_1 = x$ ; havoc  $x$ ; assume  $x = x_1 + 1$ ;  
assume  $p_1 = p$ ; havoc  $p$ ; assume  $p = p_1 + m$ ;  
assert  $p = x * m \wedge x \leq n$ ; assume false,  $p = n * m$ )  
 $\wedge$  WP (assume  $x \geq n$ ,  $p = n * m$ ))

# Example: VC Generation

Computing the weakest precondition

WP ( **assume**  $n \geq 0$ ;  
assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
assert  $p = x * m \wedge x \leq n$ ;  
havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,  
WP (assume  $x < n$ ;  
assume  $x_1 = x$ ; havoc  $x$ ; assume  $x = x_1 + 1$ ;  
assume  $p_1 = p$ ; havoc  $p$ ; assume  $p = p_1 + m$ ;  
assert  $p = x * m \wedge x \leq n$ ; assume false,  $p = n * m$ )  
 $\wedge x \geq n \Rightarrow p = n * m$ )

# Example: VC Generation

Computing the weakest precondition

**WP** ( **assume**  $n \geq 0$ ;  
assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
assert  $p = x * m \wedge x \leq n$ ;  
havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,  
**WP** (assume  $x < n$ ;  
assume  $x_1 = x$ ; havoc  $x$ ; assume  $x = x_1 + 1$ ;  
assume  $p_1 = p$ ; havoc  $p$ ; assume  $p = p_1 + m$ ;  
assert  $p = x * m \wedge x \leq n$ , **WP** ( **assume** false,  $p = n * m$ )  
 $\wedge x \geq n \Rightarrow p = n * m$ )

# Example: VC Generation

Computing the weakest precondition

**WP** ( **assume**  $n \geq 0$ ;

**assume**  $p_0 = p$ ; **havoc**  $p$ ; **assume**  $p = 0$ ;

**assume**  $x_0 = x$ ; **havoc**  $x$ ; **assume**  $x = 0$ ;

**assert**  $p = x * m \wedge x \leq n$ ;

**havoc**  $x$ ; **havoc**  $p$ ; **assume**  $p = x * m \wedge x \leq n$ ,

**WP** (**assume**  $x < n$ ;

**assume**  $x_1 = x$ ; **havoc**  $x$ ; **assume**  $x = x_1 + 1$ ;

**assume**  $p_1 = p$ ; **havoc**  $p$ ; **assume**  $p = p_1 + m$ ;

**assert**  $p = x * m \wedge x \leq n$ , **false**  $\Rightarrow p = n * m$ )

$\wedge x \geq n \Rightarrow p = n * m$ )

# Example: VC Generation

Computing the weakest precondition

**WP** ( **assume**  $n \geq 0$ ;  
assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
assert  $p = x * m \wedge x \leq n$ ;  
havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,  
**WP** (assume  $x < n$ ;  
assume  $x_1 = x$ ; havoc  $x$ ; assume  $x = x_1 + 1$ ;  
assume  $p_1 = p$ ; havoc  $p$ ; assume  $p = p_1 + m$ ;  
assert  $p = x * m \wedge x \leq n$ , **true**)  
 **$\wedge x \geq n \Rightarrow p = n * m$** )



# Example: VC Generation

Computing the weakest precondition

WP ( assume  $n \geq 0$ ;  
assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
assert  $p = x * m \wedge x \leq n$ ;  
havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,  
WP (assume  $x < n$ ;  
assume  $x_1 = x$ ; havoc  $x$ ; assume  $x = x_1 + 1$ ;  
assume  $p_1 = p$ ; havoc  $p$ ; assume  $p = p_1 + m$ ,  
 $p = x * m \wedge x \leq n$ )  
 $\wedge x \geq n \Rightarrow p = n * m$ )

# Example: VC Generation

Computing the weakest precondition

WP ( assume  $n \geq 0$ ;  
assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
assert  $p = x * m \wedge x \leq n$ ;  
havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,  
WP (assume  $x < n$ ;  
assume  $x_1 = x$ ; havoc  $x$ ; assume  $x = x_1 + 1$ ;  
assume  $p_1 = p$ ; havoc  $p$ ,  
 $p = p_1 + m \Rightarrow p = x * m \wedge x \leq n$ )  
 $\wedge x \geq n \Rightarrow p = n * m$ )

# Example: VC Generation

Computing the weakest precondition

WP ( assume  $n \geq 0$ ;  
assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
assert  $p = x * m \wedge x \leq n$ ;  
havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,  
WP (assume  $x < n$ ;  
assume  $x_1 = x$ ; havoc  $x$ ; assume  $x = x_1 + 1$ ,  
 $p_1 = p \wedge pa_1 = p_1 + m \Rightarrow pa_1 = x * m \wedge x \leq n$ )  
 $\wedge x \geq n \Rightarrow p = n * m$ )

# Example: VC Generation

Computing the weakest precondition

**WP** ( **assume**  $n \geq 0$ ;  
assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
assert  $p = x * m \wedge x \leq n$ ;  
havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,  
**WP** (assume  $x < n$ ; assume  $x_1 = x$ ; havoc  $x$ ,  
 $x = x_1 + 1 \wedge p_1 = p \wedge pa_1 = p_1 + m$   
 $\Rightarrow pa_1 = x * m \wedge x \leq n$ )  
 $\wedge x \geq n \Rightarrow p = n * m$ )

# Example: VC Generation

Computing the weakest precondition

WP ( **assume**  $n \geq 0$ ;  
assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
assert  $p = x * m \wedge x \leq n$ ;  
havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,  
WP (assume  $x < n$ ; assume  $x_1 = x$ ,  
 $xa_1 = x_1 + 1 \wedge p_1 = p \wedge pa_1 = p_1 + m$   
 $\Rightarrow pa_1 = xa_1 * m \wedge xa_1 \leq n$ )  
 $\wedge x \geq n \Rightarrow p = n * m$ )

# Example: VC Generation

Computing the weakest precondition

**WP** ( **assume**  $n \geq 0$ ;  
assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;  
assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;  
assert  $p = x * m \wedge x \leq n$ ;  
havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,  
**WP** (assume  $x < n$ ,  
 $x_1 = x \wedge xa_1 = x_1 + 1 \wedge p_1 = p \wedge pa_1 = p_1 + m$   
 $\Rightarrow pa_1 = xa_1 * m \wedge xa_1 \leq n$ )  
 $\wedge x \geq n \Rightarrow p = n * m$ )

# Example: VC Generation

Computing the weakest precondition

WP ( **assume  $n \geq 0$ ;**

assume  $p_0 = p$ ; havoc  $p$ ; assume  $p = 0$ ;

assume  $x_0 = x$ ; havoc  $x$ ; assume  $x = 0$ ;

assert  $p = x * m \wedge x \leq n$ ;

havoc  $x$ ; havoc  $p$ ; assume  $p = x * m \wedge x \leq n$ ,

$(x < n \wedge x_1 = x \wedge xa_1 = x_1 + 1 \wedge p_1 = p \wedge pa_1 = p_1 + m$

$\Rightarrow pa_1 = xa_1 * m \wedge xa_1 \leq n)$

$\wedge x \geq n \Rightarrow p = n * m)$

# Example: VC Generation

Computing the weakest precondition

$$n \geq 0 \wedge p_0 = p \wedge pa_3 = 0 \wedge x_0 = x \wedge xa_3 = 0 \Rightarrow pa_3 = xa_3 * m \wedge xa_3 \leq n \wedge$$

$$(pa_2 = xa_2 * m \wedge xa_2 \leq n \Rightarrow$$

$$((xa_2 < n \wedge x_1 = xa_2 \wedge xa_1 = x_1 + 1 \wedge$$

$$p_1 = pa_2 \wedge pa_1 = p_1 + m) \Rightarrow pa_1 = xa_1 * m \wedge$$

$$xa_1 \leq n)$$

$$\wedge (xa_2 \geq n \Rightarrow pa_2 = n * m))$$



# Example: VC Generation

The resulting VC is equivalent to the conjunction of the following implications

$$n \geq 0 \wedge p_0 = p \wedge pa_3 = 0 \wedge x_0 = x \wedge xa_3 = 0 \Rightarrow \\ pa_3 = xa_3 * m \wedge xa_3 \leq n$$

$$n \geq 0 \wedge p_0 = p \wedge pa_3 = 0 \wedge x_0 = x \wedge xa_3 = 0 \wedge pa_2 = xa_2 * m \wedge \\ xa_2 \leq n \Rightarrow$$

$$xa_2 \geq n \Rightarrow pa_2 = n * m$$

$$n \geq 0 \wedge p_0 = p \wedge pa_3 = 0 \wedge x_0 = x \wedge xa_3 = 0 \wedge pa_2 = xa_2 * m \wedge xa_2 < n \wedge \\ x_1 = xa_2 \wedge xa_1 = x_1 + 1 \wedge p_1 = pa_2 \wedge pa_1 = p_1 + m \Rightarrow \\ pa_1 = xa_1 * m \wedge xa_1 \leq n$$

# Example: VC Generation

simplifying the constraints yields

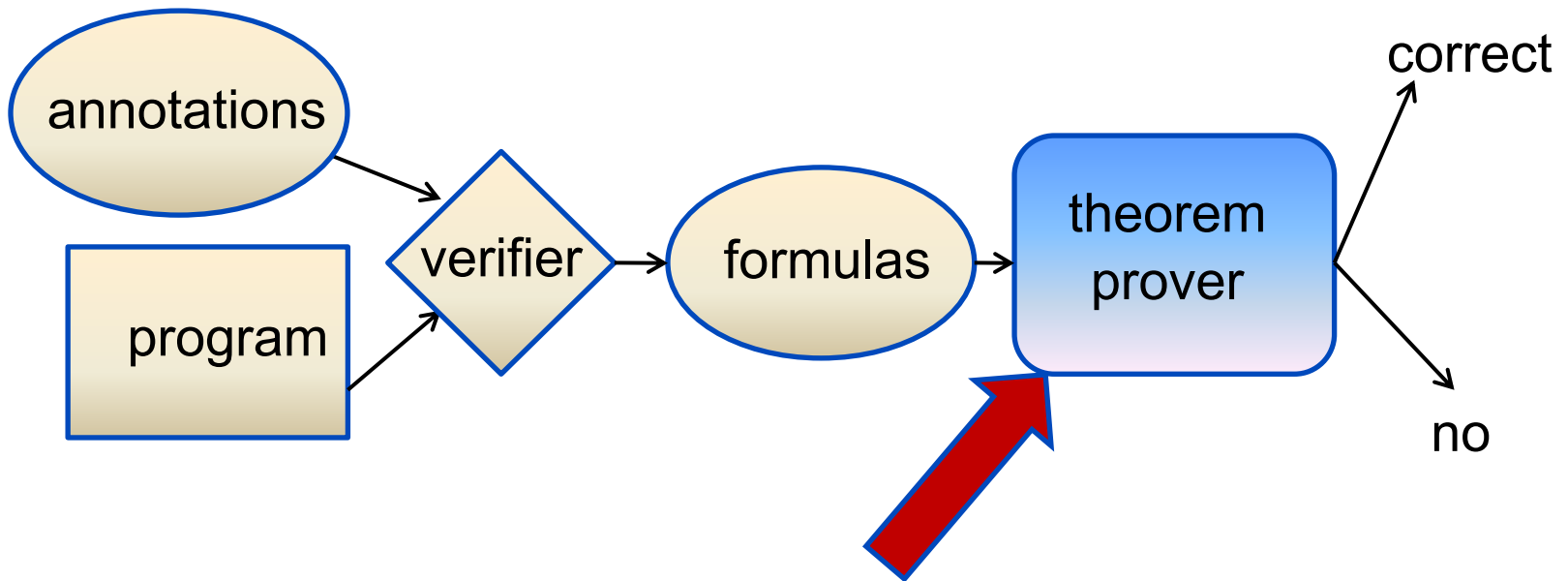
$$n \geq 0 \Rightarrow 0 = 0 * m \wedge 0 \leq n$$

$$xa_2 \leq n \wedge xa_2 \geq n \Rightarrow xa_2 * m = n * m$$

$$xa_2 < n \Rightarrow xa_2 * m + m = (xa_2 + 1) * m \wedge xa_2 + 1 \leq n$$

all of these implications are valid, which proves that the original Hoare triple was valid, too.

# Software Verification



**Prove formulas automatically!**