Verification Condition Generation

Testing, Quality Assurance, and Maintenance
Winter 2019

Prof. Arie Gurfinkel

based on slides by Prof. Ruzica Piskac and others

UNIVERSITY OF WATERLOO

Verification Condition Generation in a Nutshell

Program

{A} **P** {B}

GCGen →

loop free

Guarded Command

{A} $\mathbf{P}_{GC}$ {B}

SymEx or WP →

Verification Condition

$\mathbf{P}_{VC}$

FOL

$\mathbf{P}_{VC}$ is valid if and only if ⊢{A} **P** {B}

UNIVERSITY OF WATERLOO

Loop-Free Guarded Commands

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

```
c ::=  assume b
     | assert b
     | havoc x
     | c1 ; c2
     | c1 || c2
```

$c ::= \texttt{assume } b \mid \texttt{assert } b \mid \texttt{havoc } x \mid c_1 \texttt{ ; } c_2 \mid c_1 \texttt{ || } c_2$ *(non-deterministic choice)*

UNIVERSITY OF WATERLOO

From Programs to Guarded Commands

GC(`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(`if` b `then` c_1 `else` c_2) =

(assume b; GC(c_1)) || (assume $\neg b$; GC(c_2))

GC(`while` b inv I `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, ..., 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 $p = x * m \land x \leq n$;
havoc x; havoc p; assume $p = x * m \land x \leq n$;
(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 $p = x * m \land x \leq n$; assume false)
 assume $x \geq n$;
{ $p = n * m$ }

UNIVERSITY OF WATERLOO

Verification Condition Generation

Idea 1: Exhaustive symbolic execution of 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

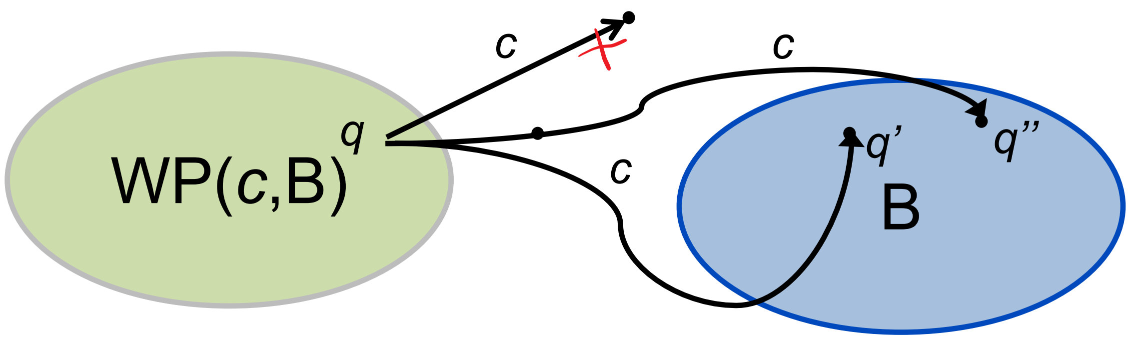
Idea 2: propagate the post-condition backwards through the program:

- From a Hoare triple {A} P {B}
- generate FOL formula A ⇒ 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:

$q \models WP(c,B)$ 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.

Computing Weakest Preconditions

WP(assume b, B) = $b \Rightarrow$ B

WP(assert b, B) = $b \wedge$ B

WP(havoc x, B) = B[a/x] (a fresh in B)

WP(c_1;c_2, B) = WP(c_1, WP(c_2, B))

WP(c_1 || c_2,B) = WP(c_1, B) $\wedge$ WP(c_2, B)

UNIVERSITY OF WATERLOO

Putting Everything Together

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

Compute c_H = assume A; GC(P); assert B

Compute $VC_H = WP(c_H, true)$

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

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

(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 $p = x * m \land x \leq n$; 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 $p = x * m \land x \leq n$;

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

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 $p = x * m \land x \leq n$; 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 $p = x * m \land x \leq n$;
havoc x; havoc p; assume $p = x * m \land x \leq n$,
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 $p = x * m \land x \leq n$; assume false, $p = n * m$)
$\land$ 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 $p = x * m \land x \leq n;$

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

$WP($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 $p = x * m \land x \leq n;$ assume false, $p = n * m)$

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

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 $n \ge 0$;

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

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

assert $p = x * m \land x \le n$;

havoc x; havoc p; assume $p = x * m \land x \le n$,

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 $p = x * m \land x \le n$, false $\Rightarrow$ $p = n * m$)

$\land\ x \ge n \Rightarrow p = n * m$)

UNIVERSITY OF WATERLOO

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 $p = x * m \land x \leq n$;

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

$WP($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 $p = x * m \land x \leq n$, true)

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

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 $p = x * m \land x \leq n$;
havoc x; havoc p; assume $p = x * m \land x \leq n$,
WP (assume $x < n$;
assume $x_1 = x$; havoc x; assume $x = x_1 + 1$;
assume $p_1 = p$; havoc p; assume $p = p_1 + m$,
$p = x * m \land x \leq n$)
$\land\ x \geq n \Rightarrow p = n * m$)

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 $p = x * m \land x \leq n$;

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

WP (assume $x < n$;

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

assume $p_1 = p$; havoc p,

$p = p_1 + m \Rightarrow p = x * m \land x \leq n$)

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

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 $p = x * m \land x \leq n$;

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

WP (assume $x < n$;

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

$p_1 = p \land pa_1 = p_1 + m \Rightarrow pa_1 = x * m \land x \leq n$)

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

UNIVERSITY OF WATERLOO

Example: VC Generation

Computing the weakest precondition

$$
\begin{aligned}
\mathrm{WP}\ (\quad & \text{assume } n \geq 0; \\
& \text{assume } p_0 = p;\ \text{havoc } p;\ \text{assume } p = 0; \\
& \text{assume } x_0 = x;\ \text{havoc } x;\ \text{assume } x = 0; \\
& \text{assert } p = x * m \wedge x \leq n; \\
& \text{havoc } x;\ \text{havoc } p;\ \text{assume } p = x * m \wedge x \leq n, \\
& \mathrm{WP}\ (\text{assume } x < n;\ \text{assume } x_1 = x;\ \text{havoc } x, \\
& \quad x = x_1 + 1 \wedge p_1 = p \wedge pa_1 = p_1 + m \\
& \quad\quad \Rightarrow pa_1 = x * m \wedge x \leq n) \\
& \wedge x \geq n \Rightarrow p = n * m)
\end{aligned}
$$

Example: VC Generation

Computing the weakest precondition

$$
\begin{aligned}
WP(\quad & \text{assume } n \geq 0; \\
& \text{assume } p_0 = p; \text{ havoc } p; \text{ assume } p = 0; \\
& \text{assume } x_0 = x; \text{ havoc } x; \text{ assume } x = 0; \\
& \text{assert } p = x * m \land x \leq n; \\
& \text{havoc } x; \text{ havoc } p; \text{ assume } p = x * m \land x \leq n, \\
& \; WP(\text{assume } x < n; \text{ assume } x_1 = x, \\
& \quad xa_1 = x_1 + 1 \land p_1 = p \land pa_1 = p_1 + m \\
& \quad\quad \Rightarrow pa_1 = xa_1 * m \land xa_1 \leq n) \\
& \land x \geq n \Rightarrow p = n * m)
\end{aligned}
$$

UNIVERSITY OF WATERLOO

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 $p = x * m \land x \leq n;$

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

$WP($assume $x < n,$

$x_1 = x \land xa_1 = x_1 + 1 \land p_1 = p \land pa_1 = p_1 + m$

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

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

UNIVERSITY OF WATERLOO

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

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

UNIVERSITY OF WATERLOO

Example: VC Generation

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

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

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

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

UNIVERSITY OF WATERLOO

Example: VC Generation

simplifying the constraints yields

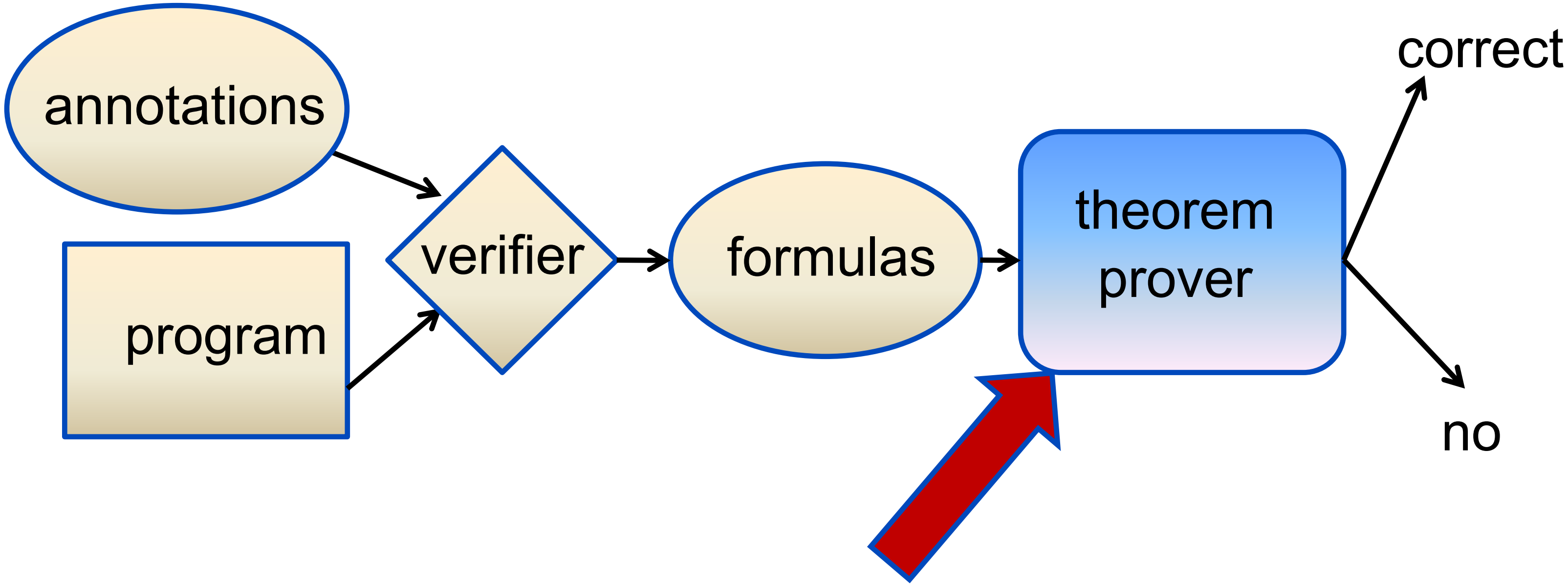
$n \geq 0 \Rightarrow 0 = 0 * m \wedge 0 \leq n$

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

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

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

Software Verification



Prove formulas automatically!