STRONGEST POST-CONDITION

SYMBOLIC EXECUTION IN THE FORWARD DIRECTION

- Given a set of states S and a command c, the strongest post-condition $sp(S, c)$ consists of all states that can be obtained after executing c on any state in S.

$$sp(S, c) \triangleq \{\sigma' \mid \exists \sigma \in S \,.\, (\sigma, c) \hookrightarrow^* (\sigma', \mathsf{skip})\}$$

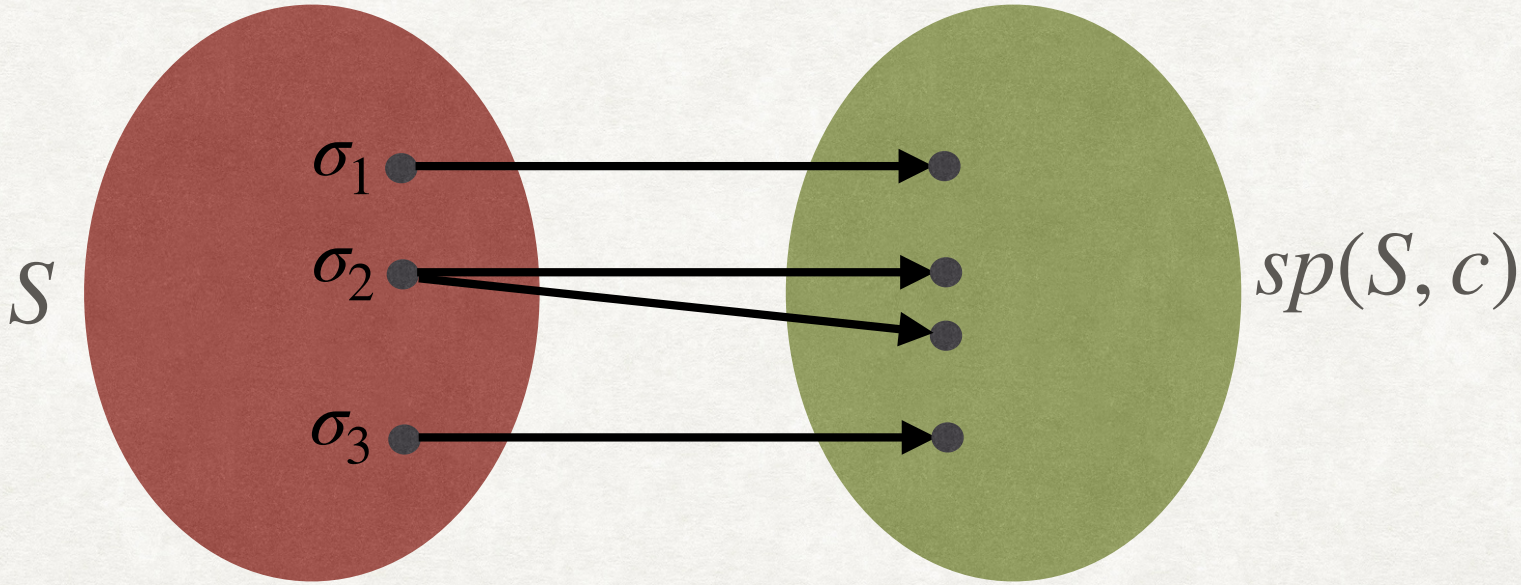
Equivalently, $\sigma' \in sp(S, c) \leftrightarrow \exists \sigma \in S \,.\, (\sigma, c) \hookrightarrow^* (\sigma', skip)$

STRONGEST POST-CONDITION

SYMBOLIC EXECUTION IN THE FORWARD DIRECTION

- Given a set of states S and a command c, the strongest post-condition $sp(S, c)$ consists of all states that can be obtained after executing c on any state in S.

$$sp(S, c) \triangleq \{\sigma' \mid \exists \sigma \in S \,.\, (\sigma, c) \hookrightarrow^* (\sigma', \mathsf{skip})\}$$

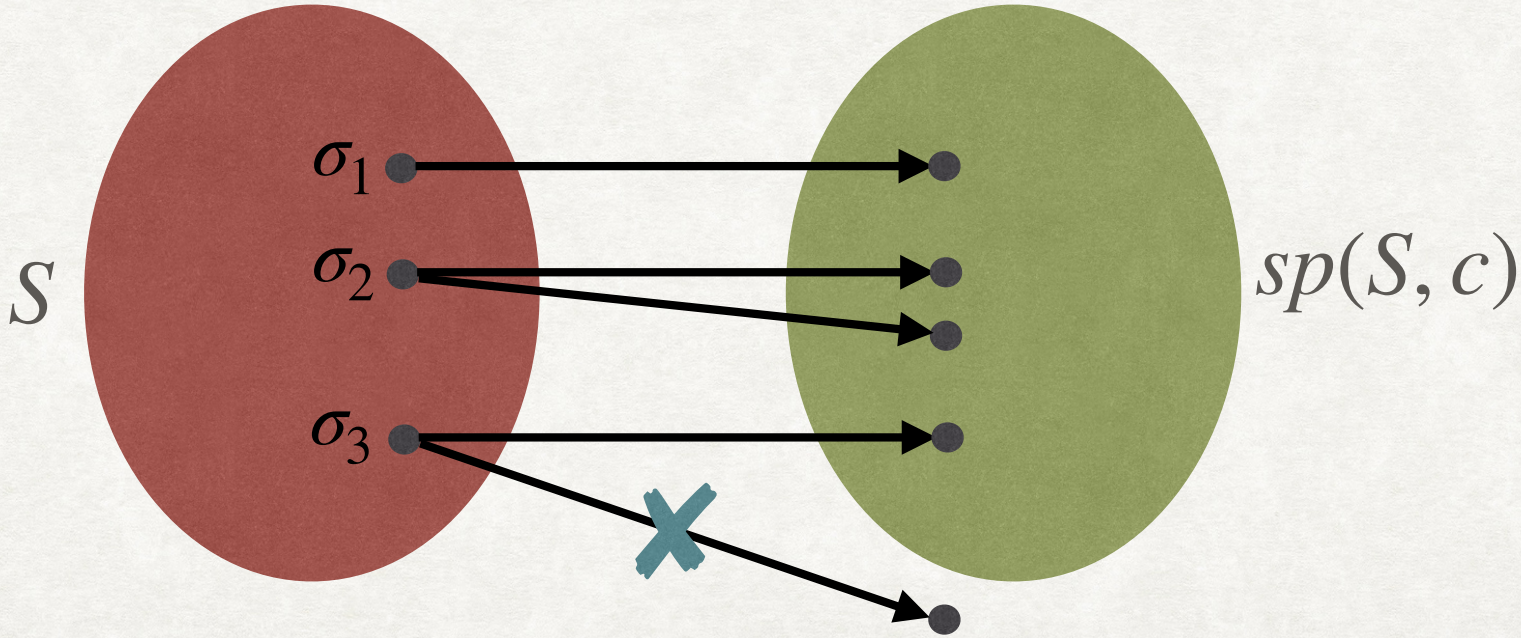


STRONGEST POST-CONDITION

SYMBOLIC EXECUTION IN THE FORWARD DIRECTION

- Given a set of states S and a command c, the strongest post-condition $sp(S, c)$ consists of all states that can be obtained after executing c on any state in S.

$$sp(S, c) \triangleq \{\sigma' \mid \exists \sigma \in S \,.\, (\sigma, c) \hookrightarrow^* (\sigma', \mathsf{skip})\}$$

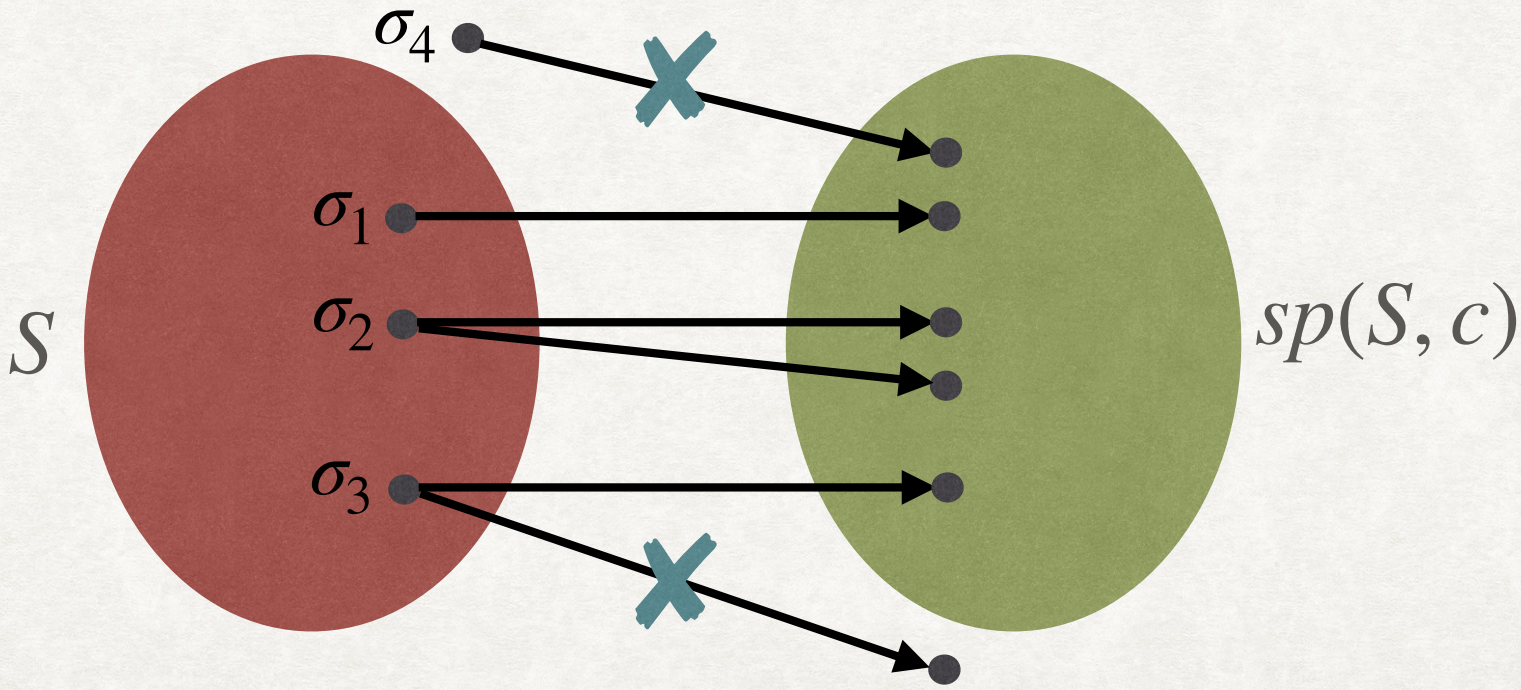


STRONGEST POST-CONDITION

SYMBOLIC EXECUTION IN THE FORWARD DIRECTION

- Given a set of states S and a command c, the strongest post-condition $sp(S, c)$ consists of all states that can be obtained after executing c on any state in S.

$$sp(S, c) \triangleq \{\sigma' \mid \exists \sigma \in S \,.\, (\sigma, c) \hookrightarrow^* (\sigma', \mathsf{skip})\}$$

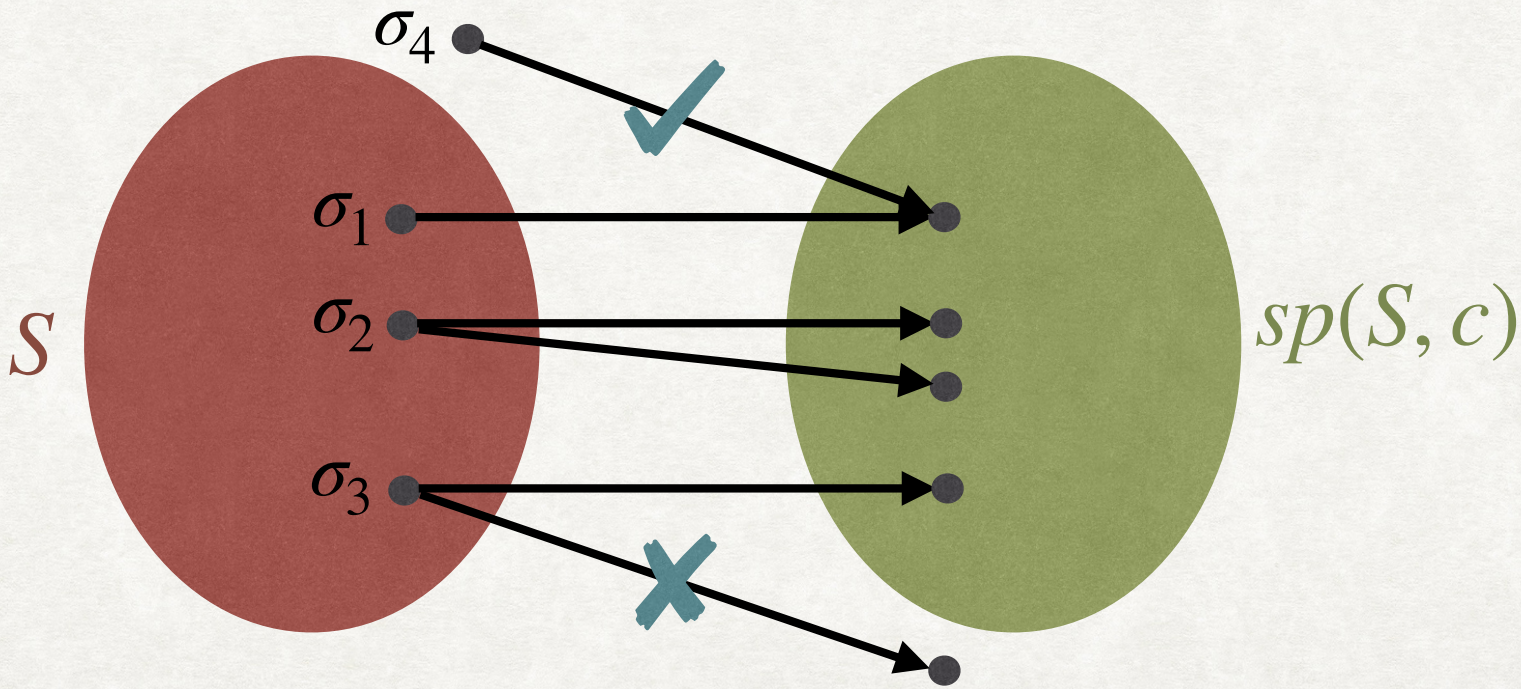


STRONGEST POST-CONDITION

SYMBOLIC EXECUTION IN THE FORWARD DIRECTION

- Given a set of states S and a command c, the strongest post-condition $sp(S, c)$ consists of all states that can be obtained after executing c on any state in S.

$$sp(S, c) \triangleq \{\sigma' \mid \exists \sigma \in S \,.\, (\sigma, c) \hookrightarrow^* (\sigma', \mathsf{skip})\}$$



STRONGEST POST-CONDITION

SYMBOLIC EXECUTION IN THE FORWARD DIRECTION

- Given a set of states S and a command c, the strongest post-condition $sp(S, c)$ consists of all states that can be obtained after executing c on any state in S.

$$sp(S, c) \triangleq \{\sigma' \mid \exists \sigma \in S \,.\, (\sigma, c) \hookrightarrow^* (\sigma', \mathsf{skip})\}$$

- We can use a FOL formula F to represent a set of states.
- The symbolic strongest post-condition operator can be defined as:

$$\sigma' \vDash sp(F, \mathsf{c}) \Leftrightarrow \exists \sigma \,.\, \sigma \vDash F \wedge (\sigma, \mathsf{c}) \hookrightarrow^* (\sigma', \mathsf{skip})$$

- We can now use the semantics in FOL (ρ) to define symbolic sp:

$$sp(F, \mathsf{c}) \triangleq (\exists V \,.\, F \wedge \rho(\mathsf{c}))[V/V']$$

FIRST ELIMINATE EXISTENTIAL QUANTIFICATION ON V, THEN SUBSTITUTE V FOR V'

QUANTIFIER ELIMINATION

- Eliminate quantifiers in a formula F to obtain an equivalent formula G (equivalent modulo $T_{\mathbb{Q}}$).
  - A decidable procedure exists for $T_{\mathbb{Q}}$−formulae.
  - Ferrante and Rackoff's Method (BM Chapter 7)
- Consider the formula: $\exists y \,.\, x = y + 1$.
  - Equivalent formula after eliminating y: $\top$
- Consider the formula: $\exists y \,.\, y > 1 \wedge x = 2y$
  - Equivalent formula after eliminating y: $x > 2$
- What about $\exists y \,.\, x = 2y \wedge x > y$?
  - Equivalent formula: $x > 0$

STRONGEST POST-CONDITION

EXAMPLE

$$sp(F, \mathsf{c}) \triangleq (\exists V \,.\, F \wedge \rho(\mathsf{c}))[V/V']$$

Lets calculate $sp(\mathsf{y} > 0, \mathsf{x}:=\mathsf{y}+1)$

STRONGEST POST-CONDITION

EXAMPLE

$$sp(F, \mathsf{c}) \triangleq (\exists V \,.\, F \wedge \rho(\mathsf{c}))[V/V']$$

Lets calculate $sp(\mathsf{y} > 0, \mathsf{x}{:=}\mathsf{y}{+}1)$

$$\begin{aligned} sp(\mathsf{y} > 0, \mathsf{x}{:=}\mathsf{y}{+}1) &\triangleq \exists \mathsf{x} \,.\, \exists \mathsf{y} \,.\, \mathsf{y} > 0 \wedge \rho(\mathsf{x}{:=}\mathsf{y}{+}1) \\ &\equiv \exists \mathsf{x} \,.\, \exists \mathsf{y} \,.\, \mathsf{y} > 0 \wedge \mathsf{x}' = \mathsf{y} + 1 \wedge \mathsf{y}' = \mathsf{y} \\ &\equiv \mathsf{y}' > 0 \wedge \mathsf{x}' = \mathsf{y}' + 1 \\ &\equiv \mathsf{y} > 0 \wedge \mathsf{x} = \mathsf{y} + 1 \end{aligned}$$

Eliminate x and y

Substitute x′ and y′ with x and y

ANNOUNCEMENT

- Assignments : Late Submission Policy
  - 1 Day late : 25% Penalty
  - 2 Days late : 50% Penalty
  - Submissions after 2 days will be ignored.

STRONGEST POST-CONDITION

EXAMPLE

$$sp(F, \mathsf{c}) \triangleq (\exists V \,.\, F \wedge \rho(\mathsf{c}))[V/V']$$

Lets calculate $sp(\mathsf{y} > 0, \mathsf{x}:=\mathsf{y}+1)$

$$\begin{aligned} sp(\mathsf{y} > 0, \mathsf{x}:=\mathsf{y}+1) &\triangleq \exists \mathsf{x} \,.\, \exists \mathsf{y} \,.\, \mathsf{y} > 0 \wedge \rho(\mathsf{x}:=\mathsf{y}+1) \\ &\equiv \exists \mathsf{x} \,.\, \exists \mathsf{y} \,.\, \mathsf{y} > 0 \wedge \mathsf{x}' = \mathsf{y} + 1 \wedge \mathsf{y}' = \mathsf{y} \\ &\equiv \mathsf{y}' > 0 \wedge \mathsf{x}' = \mathsf{y}' + 1 \\ &\equiv \mathsf{y} > 0 \wedge \mathsf{x} = \mathsf{y} + 1 \end{aligned}$$

Alternative Formulation for Assignment Statement:

$$sp(F, \mathsf{x}:=\mathsf{e}) \equiv \exists x' \,.\, F[x'/x] \wedge x = e[x'/x]$$

STRONGEST POST-CONDITION

MORE EXAMPLES

$sp(\mathsf{y} > 0, \mathsf{x}:=\mathsf{havoc}) \triangleq ???$

STRONGEST POST-CONDITION

MORE EXAMPLES

$$sp(y > 0, x:=\text{havoc}) \triangleq \exists x\,.\,\exists y\,.\; y > 0 \wedge y' = y \quad [\rho(x:=\text{havoc}) \triangleq frame(x)]$$

$$\triangleq y > 0$$

STRONGEST POST-CONDITION

MORE EXAMPLES

$$sp(\mathsf{y} > 0, \mathsf{x}:=\mathsf{havoc}) \triangleq \exists \mathsf{x} \,.\, \exists \mathsf{y} \,.\, \mathsf{y} > 0 \wedge \mathsf{y}' = \mathsf{y} \quad [\rho(\mathsf{x}:=\mathsf{havoc}) \triangleq frame(\mathsf{x})]$$

$$\triangleq \mathsf{y} > 0$$

$$sp(F, \mathsf{x}:=\mathsf{havoc}) \triangleq \exists \mathsf{x} \,.\, F$$

STRONGEST POST-CONDITION

MORE EXAMPLES

$$sp(\mathsf{y} > 0, \mathsf{x}:=\mathsf{havoc}) \triangleq \exists \mathsf{x} \,.\, \exists \mathsf{y} \,.\, \mathsf{y} > 0 \wedge \mathsf{y}' = \mathsf{y}$$
$$\triangleq \mathsf{y} > 0$$

$$sp(\mathsf{F}, \mathsf{assume}(\mathsf{G})) \triangleq ???$$

STRONGEST POST-CONDITION

MORE EXAMPLES

$$sp(y > 0, x:=\text{havoc}) \triangleq \exists x\,.\,\exists y\,.\; y > 0 \wedge y' = y$$
$$\triangleq y > 0$$

$$sp(F, \text{assume}(G)) \triangleq F \wedge G$$

STRONGEST POST-CONDITION

MORE EXAMPLES

$sp(\mathsf{y} > 0, \mathsf{x}:=\mathsf{havoc}) \triangleq \exists \mathsf{x} \,.\, \exists \mathsf{y} \,.\, \mathsf{y} > 0 \wedge \mathsf{y}' = \mathsf{y}$

$\triangleq \mathsf{y} > 0$

$sp(\mathsf{F}, \mathsf{assume}(\mathsf{G})) \triangleq \mathsf{F} \wedge \mathsf{G}$

$sp(\mathsf{F}, \mathsf{assert}(\mathsf{G})) \triangleq ???$

STRONGEST POST-CONDITION

MORE EXAMPLES

$$sp(\mathsf{y} > 0, \mathsf{x{:=}havoc}) \triangleq \exists \mathsf{x} \,.\, \exists \mathsf{y} \,.\, \mathsf{y} > 0 \wedge \mathsf{y}' = \mathsf{y}$$
$$\triangleq \mathsf{y} > 0$$

$$sp(\mathsf{F}, \mathsf{assume(G)}) \triangleq \mathsf{F} \wedge \mathsf{G}$$

$$\begin{aligned} sp(\mathsf{F}, \mathsf{assert(G)}) &\triangleq \exists V \,.\, F \wedge (G \rightarrow frame(\varnothing)) \\ &\equiv \exists V \,.\, F \wedge (\neg G \vee frame(\varnothing)) \\ &\equiv \exists V \,.\, (F \wedge \neg G) \vee \exists V \,.\, (F \wedge frame(\varnothing)) \\ &\equiv \exists V \,.\, (F \wedge \neg G) \vee F[V'/V] \\ &\equiv (\exists V \,.\, F \wedge \neg G) \vee F \end{aligned}$$

STRONGEST POST-CONDITION

MORE EXAMPLES

$$sp(\mathsf{y} > 0, \mathsf{x}:=\mathsf{havoc}) \triangleq \exists \mathsf{x} \,.\, \exists \mathsf{y} \,.\, \mathsf{y} > 0 \wedge \mathsf{y}' = \mathsf{y}$$
$$\triangleq \mathsf{y} > 0$$

$$sp(\mathsf{F}, \mathsf{assume}(\mathsf{G})) \triangleq \mathsf{F} \wedge \mathsf{G}$$

$$sp(\mathsf{F}, \mathsf{assert}(\mathsf{G})) \triangleq (\exists V \,.\, \mathsf{F} \wedge \neg \mathsf{G}) \vee \mathsf{F}$$

STRONGEST POST-CONDITION

MORE EXAMPLES

$$sp(\mathsf{y} > 0, \mathsf{x{:=}havoc}) \triangleq \exists \mathsf{x} \,.\, \exists \mathsf{y} \,.\, \mathsf{y} > 0 \wedge \mathsf{y}' = \mathsf{y}$$
$$\triangleq \mathsf{y} > 0$$

$$sp(\mathsf{F}, \mathsf{assume(G)}) \triangleq \mathsf{F} \wedge \mathsf{G}$$

$$sp(\mathsf{F}, \mathsf{assert(G)}) \triangleq (\exists V \,.\, \mathsf{F} \wedge \neg \mathsf{G}) \vee \mathsf{F}$$

$$sp(\mathit{false}, \mathsf{c}) \triangleq ???$$

STRONGEST POST-CONDITION

EXAMPLES

$$sp(\mathsf{y} > 0, \mathsf{x{:=}havoc}) \triangleq \exists \mathsf{x} \,.\, \exists \mathsf{y} \,.\, \mathsf{y} > 0 \wedge \mathsf{y}' = \mathsf{y}$$
$$\triangleq \mathsf{y} > 0$$

$$sp(\mathsf{F}, \mathsf{assume(G)}) \triangleq \mathsf{F} \wedge \mathsf{G}$$

$$sp(\mathsf{F}, \mathsf{assert(G)}) \triangleq (\exists V \,.\, \mathsf{F} \wedge \neg \mathsf{G}) \vee \mathsf{F}$$

$$sp(\mathit{false}, \mathsf{c}) \triangleq \mathit{false}$$

EXAMPLES

- $sp(x > 5, \text{assume}(x < 20)) \equiv ???$
- $sp(x > 5, \text{assert}(x < 0)) \equiv ???$
- $sp(x > 0, x := x+1) \equiv ???$

EXAMPLES

- $sp(x > 5, \text{assume}(x < 20)) \equiv x > 5 \wedge x < 20$
- $sp(x > 5, \text{assert}(x < 0)) \equiv \mathit{true}$
- $sp(x > 0, x{:=}x+1) \equiv x > 1$

STRONGEST POST-CONDITION

COMPOUND STATEMENTS

- $sp(\mathsf{F}, \mathsf{c};\mathsf{c}') \triangleq ???$

STRONGEST POST-CONDITION

COMPOUND STATEMENTS

- $sp(\mathsf{F}, \mathsf{c};\mathsf{c}') \triangleq sp(sp(\mathsf{F}, \mathsf{c}), \mathsf{c}')$ (Homework: Prove this formally.)

STRONGEST POST-CONDITION

COMPOUND STATEMENTS

- $sp(\mathsf{F}, \mathsf{c};\mathsf{c}') \triangleq sp(sp(\mathsf{F}, \mathsf{c}), \mathsf{c}')$
- $sp(\mathsf{F}, \text{if(G) then c else c}') \triangleq ???$

STRONGEST POST-CONDITION

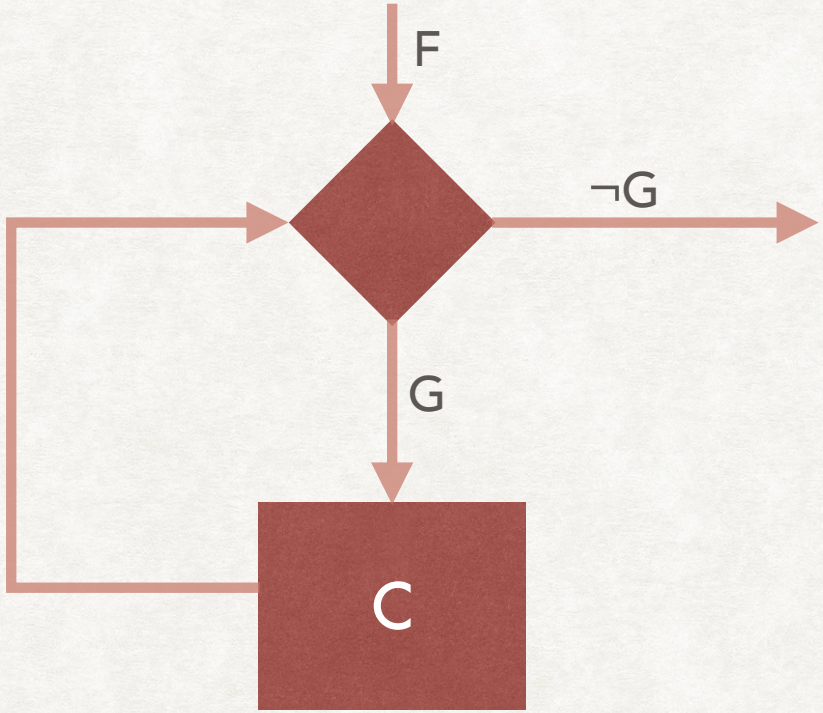
COMPOUND STATEMENTS

- $sp(\mathsf{F}, \mathsf{c};\mathsf{c}') \triangleq sp(sp(\mathsf{F}, \mathsf{c}), \mathsf{c}')$
- $sp(\mathsf{F}, \text{if}(\mathsf{G}) \text{ then } \mathsf{c} \text{ else } \mathsf{c}') \triangleq sp(\mathsf{F} \wedge \mathsf{G}, \mathsf{c}) \vee sp(\mathsf{F} \wedge \neg\mathsf{G}, \mathsf{c}')$ (Homework: Prove this formally.)

STRONGEST POST-CONDITION

WHILE LOOPS

- How to find sp(F, while(G) do c)?

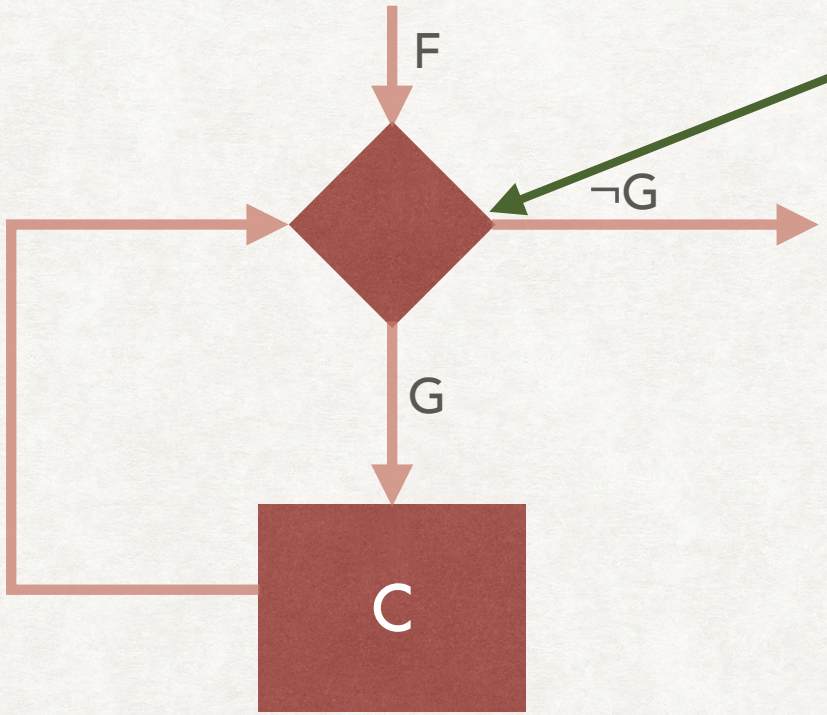


STRONGEST POST-CONDITION

WHILE LOOPS

- How to find $sp(\mathrm{F}, \text{while(G) do c})$?

Let us collect all states possible at the end of any iteration



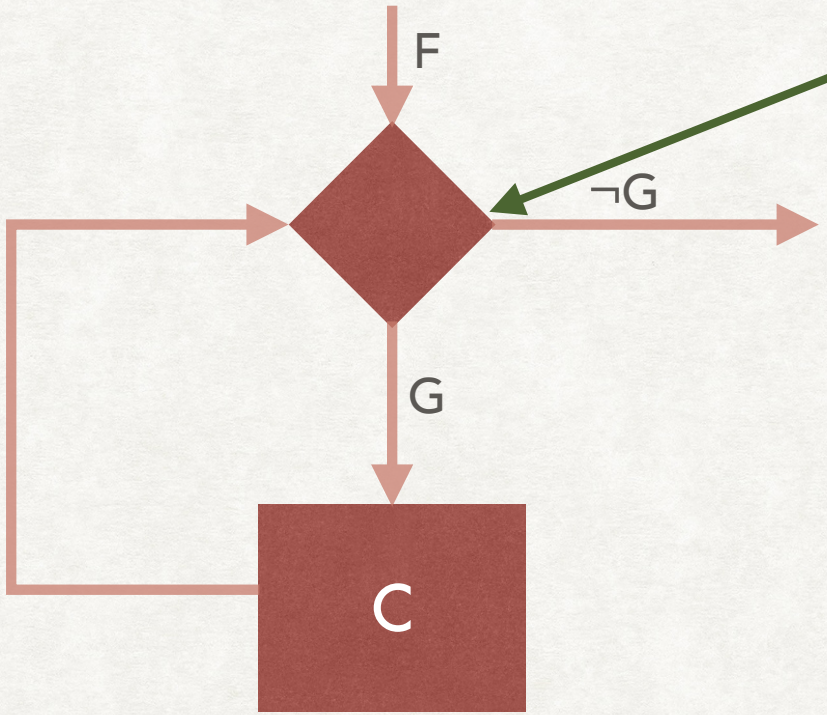
Iteration i	States possible at iteration i
0	
1	
2	
…	…

STRONGEST POST-CONDITION

WHILE LOOPS

- How to find $sp(\mathsf{F}, \text{while}(\mathsf{G}) \text{ do } \mathsf{c})$?

Let us collect all states possible at the end of any iteration

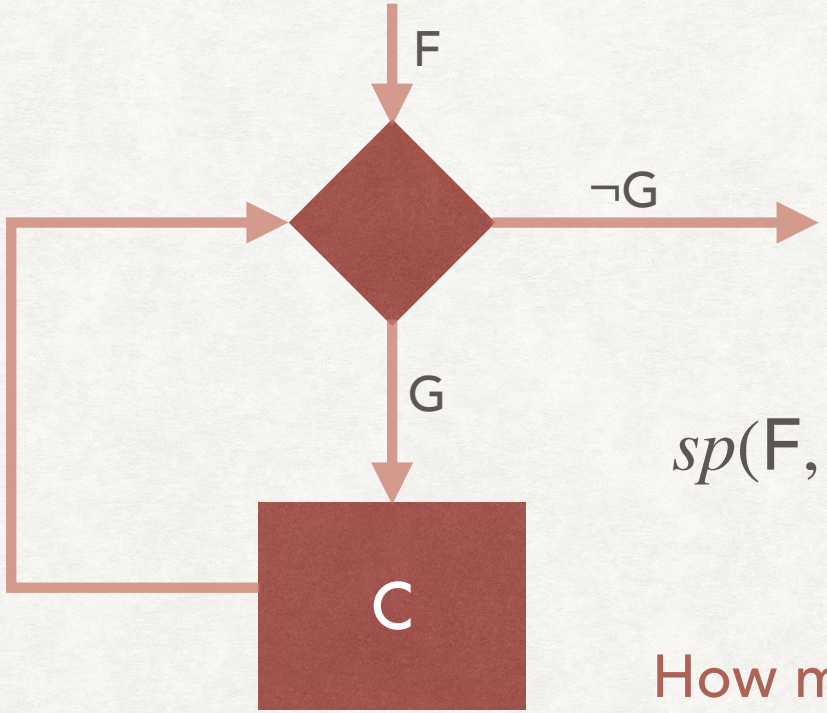


Iteration i	States possible at iteration i
0	F
1	$sp(\mathsf{F} \wedge \mathsf{G}, \mathsf{c})$
2	$sp(sp(\mathsf{F} \wedge \mathsf{G}, \mathsf{c}) \wedge \mathsf{G}, \mathsf{c})$
...	...

STRONGEST POST-CONDITION

WHILE LOOPS

- How to find $sp(\mathsf{F}, \text{while(G) do c})$?



$$F^0 = \mathsf{F}$$

$$F^k = sp(F^{k-1} \wedge \mathsf{G}, \mathsf{c})$$

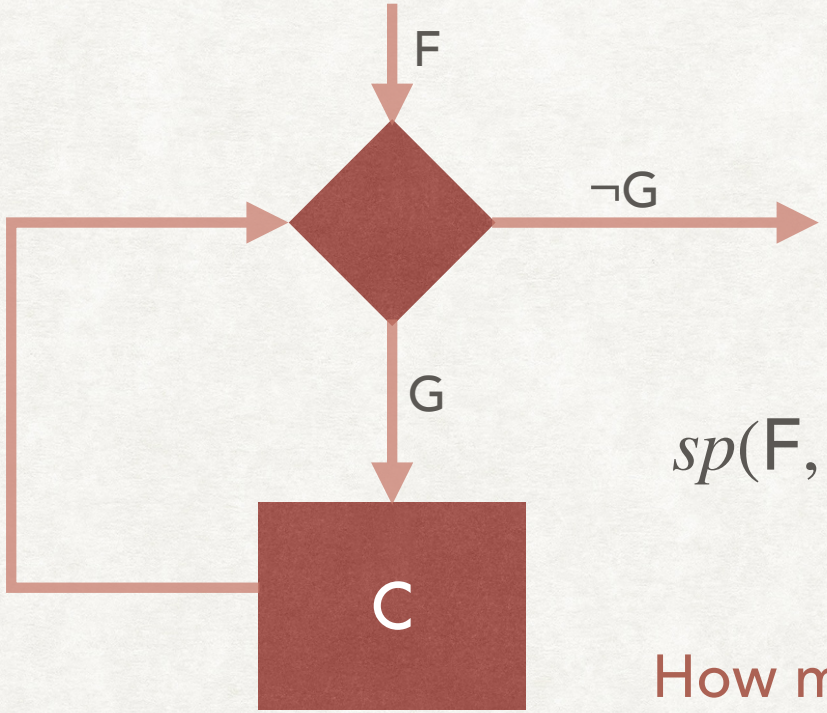
$$sp(\mathsf{F}, \text{while(G) do c}) \triangleq \bigvee_{k=0}^{\infty} F^k \wedge \neg \mathsf{G}$$

How many F^k should be calculated?

STRONGEST POST-CONDITION

WHILE LOOPS

- How to find sp(F, while(G) do c)?



$$F^0 = \mathsf{F}$$

$$F^k = sp(F^{k-1} \wedge \mathsf{G}, \mathsf{c})$$

$$sp(\mathsf{F}, \text{while(G) do c}) \triangleq \bigvee_{k=0}^{\infty} F^k \wedge \neg \mathsf{G}$$

How many F^k should be calculated?

Until $F^k \rightarrow \bigvee_{i=0}^{k-1} F^i$

WHILE LOOPS

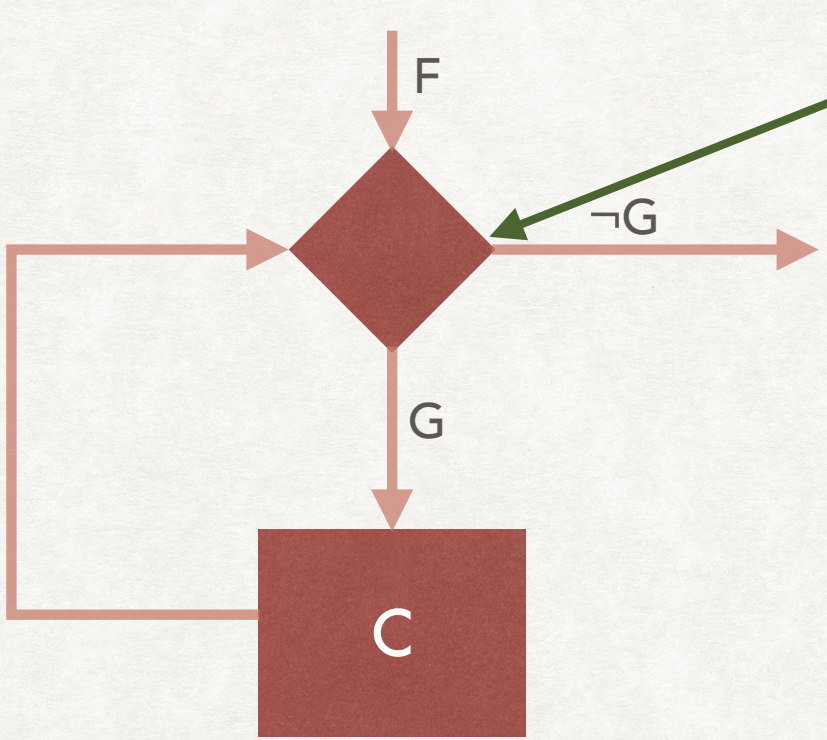
EXAMPLES

- $sp(x < 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq ???$

WHILE LOOPS

EXAMPLES

- $sp(x < 0, \text{while}(x<10)\ \text{do}\ x:=x+1;) \triangleq ???$

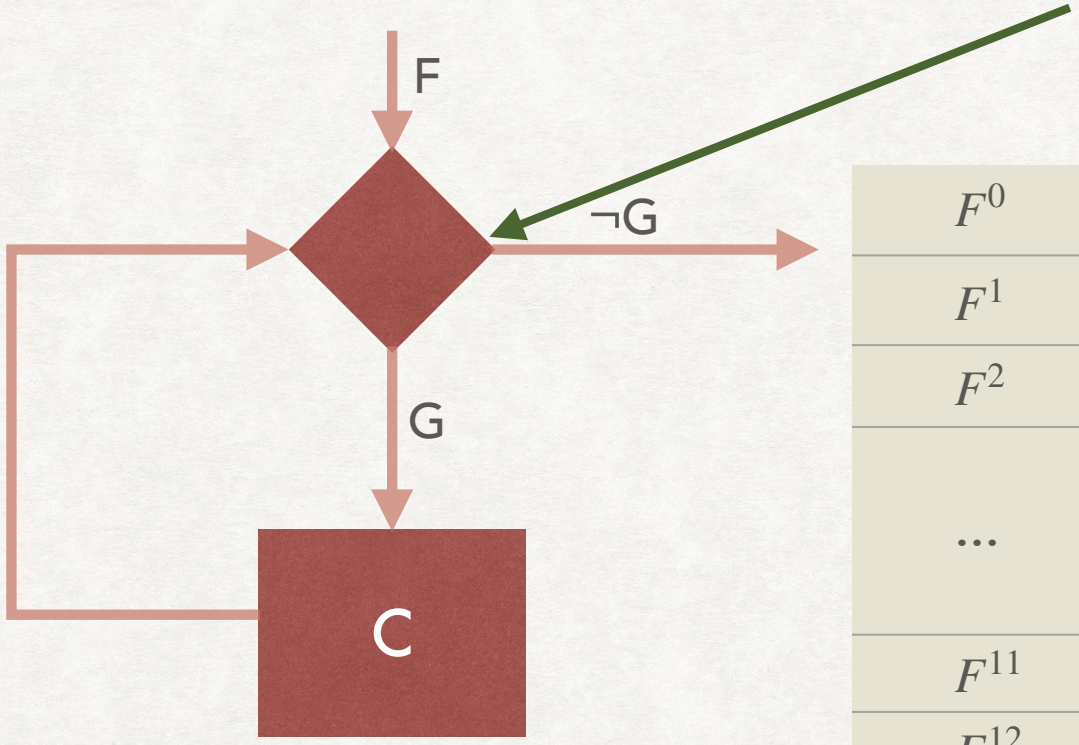


F^0	F
F^1	$sp(F \wedge G, c)$
F^2	$sp(sp(F \wedge G, c) \wedge G, c)$
...	...

WHILE LOOPS

EXAMPLES

- $sp(x < 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq ???$

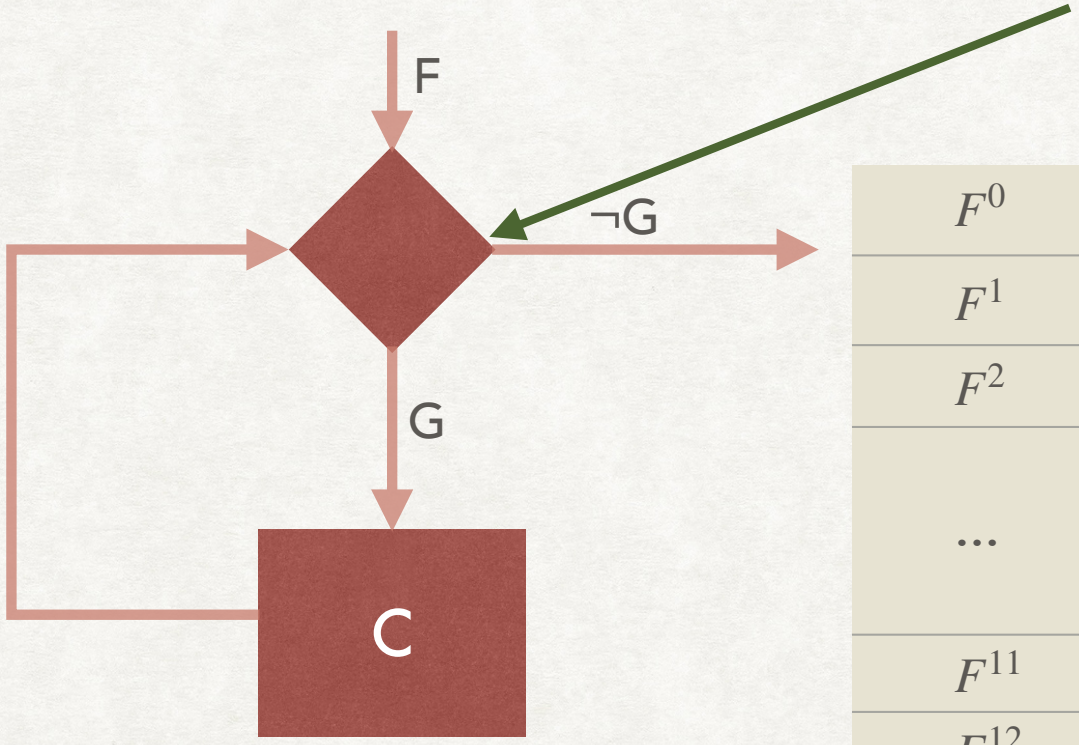


F^0	
F^1	
F^2	
...	...
F^{11}	
F^{12}	

WHILE LOOPS

EXAMPLES

- $sp(x < 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq ???$



F^0	$x < 0$
F^1	$x < 1$
F^2	$x < 2$
...	...
F^{11}	$x < 11$
F^{12}	$x < 11$

WHILE LOOPS

EXAMPLES

- $sp(x < 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq x \geq 10 \wedge x < 11$

WHILE LOOPS

EXAMPLES

- $sp(x < 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq x \geq 10 \wedge x < 11$
- $sp(x > 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq ???$

WHILE LOOPS

EXAMPLES

- $sp(x < 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq x \geq 10 \wedge x < 11$
- $sp(x > 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq x \geq 10$

WHILE LOOPS

EXAMPLES

- $sp(x < 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq x \geq 10 \wedge x < 11$
- $sp(x > 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq x \geq 10$
- $sp(x > 0, \text{while}(x>0) \text{ do } x:=x+1;) \triangleq ???$

WHILE LOOPS

EXAMPLES

- $sp(x < 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq x \geq 10 \wedge x < 11$
- $sp(x > 0, \text{while}(x<10) \text{ do } x:=x+1;) \triangleq x \geq 10$
- $sp(x > 0, \text{while}(x>0) \text{ do } x:=x+1;) \triangleq \bot$

STRONGEST POST-CONDITION AND VERIFICATION

- Given a program c with assertions, then c is safe if $sp(\text{error} = 0, \text{c}) \rightarrow \text{error} = 0$ is valid.
- The Hoare Triple $\{P\}\text{c}\{Q\}$ is valid if $sp(P, \text{c}) \rightarrow Q$ is valid.
- Do we have a decidable procedure for sp?
  - No, due to the (potentially) unbounded computation required for while loops.
  - What is an example of F and c such that $sp(F, c)$ requires unbounded computation?

COMPLETE EXAMPLE

```
1: assume(i = 0 ∧ n ≥ 0);
2: while(i < n) do
3:   i := i + 1;
4: assert(i = n);
```

$$sp(error = 0,1 :) \equiv error = 0 \wedge i = 0 \wedge n \geq 0$$

$$sp(error = 0 \wedge i = 0 \wedge n \geq 0 \wedge i < n,3 :) \equiv \exists i, n, error \,.\, error = 0 \wedge i = 0 \wedge n \geq 0 \wedge i < n \wedge i' = i + 1 \wedge frame(i)$$

COMPLETE EXAMPLE

```
1: assume(i = 0 ∧ n ≥ 0);
2: while(i < n) do
3:   i := i + 1;
4: assert(i = n);
```

$$sp(error = 0, 1 :) \equiv error = 0 \wedge i = 0 \wedge n \geq 0$$

$$sp(error = 0 \wedge i = 0 \wedge n \geq 0 \wedge i < n, 3 :) \equiv \exists i, n, error \,.\, error = 0 \wedge i = 0 \wedge n \geq 0 \wedge i < n \wedge i' = i + 1 \wedge frame(i)$$

$$\equiv error = 0 \wedge i = 1 \wedge n > 0$$

$$sp(error = 0 \wedge i = 1 \wedge n \geq 1 \wedge i < n, 3 :) \equiv error = 0 \wedge i = 2 \wedge n \geq 2$$

$$\vdots$$

COMPLETE EXAMPLE

```
1: assume(i = 0 ∧ n ≥ 0);
2: while(i < n) do
3:   i := i + 1;
4: assert(i = n);
```

$$sp(error = 0,1 :) \equiv error = 0 \wedge i = 0 \wedge n \geq 0$$

$$sp(error = 0 \wedge i = 0 \wedge n \geq 0,2 :) \equiv error = 0 \wedge ((i = 0 \wedge n \geq 0) \vee (i = 1 \wedge n \geq 1) \vee (i = 2 \wedge n \geq 2) \vee \ldots) \wedge i \geq n$$

COMPLETE EXAMPLE

```
1: assume(i = 0 ∧ n ≥ 0);
2: while(i < n) do
3:   i := i + 1;
4: assert(i = n);
```

$$sp(error = 0,1 :) \equiv error = 0 \wedge i = 0 \wedge n \geq 0$$

$$\begin{aligned} sp(error = 0 \wedge i = 0 \wedge n \geq 0,2 :) \equiv\ & error = 0 \wedge ((i = 0 \wedge n \geq 0) \\ & \vee (i = 1 \wedge n \geq 1) \vee (i = 2 \wedge n \geq 2) \vee \ldots) \\ & \wedge i \geq n \\ \equiv\ & error = 0 \wedge i = n \end{aligned}$$

$$\begin{aligned} sp(error = 0 \wedge i = n,4 :) &\equiv error = 0 \wedge i = n \vee \exists V . error = 0 \wedge i = n \wedge i \neq n \\ &\equiv error = 0 \wedge i = n \\ &\Rightarrow error = 0 \end{aligned}$$

SOUNDNESS AND COMPLETENESS

- Verification Problem: For a given program c, is $(Error, c')$ reachable for some c′?
  - A program c is safe if $(Error, c')$ is not reachable for any c′.
  - Denoted as $\vdash$ c is safe.
- Strongest Post-condition based Verification Procedure
  - c is safe if $sp(\text{error} = 0, c) \Rightarrow \text{error} = 0$.
  - Denoted as $\models$ c is safe.
- Soundness: Does $\models$ c is safe $\Rightarrow \vdash$ c is safe?
- Completeness: Does $\vdash$ c is safe $\Rightarrow \models$ c is safe?

SOUNDNESS AND COMPLETENESS

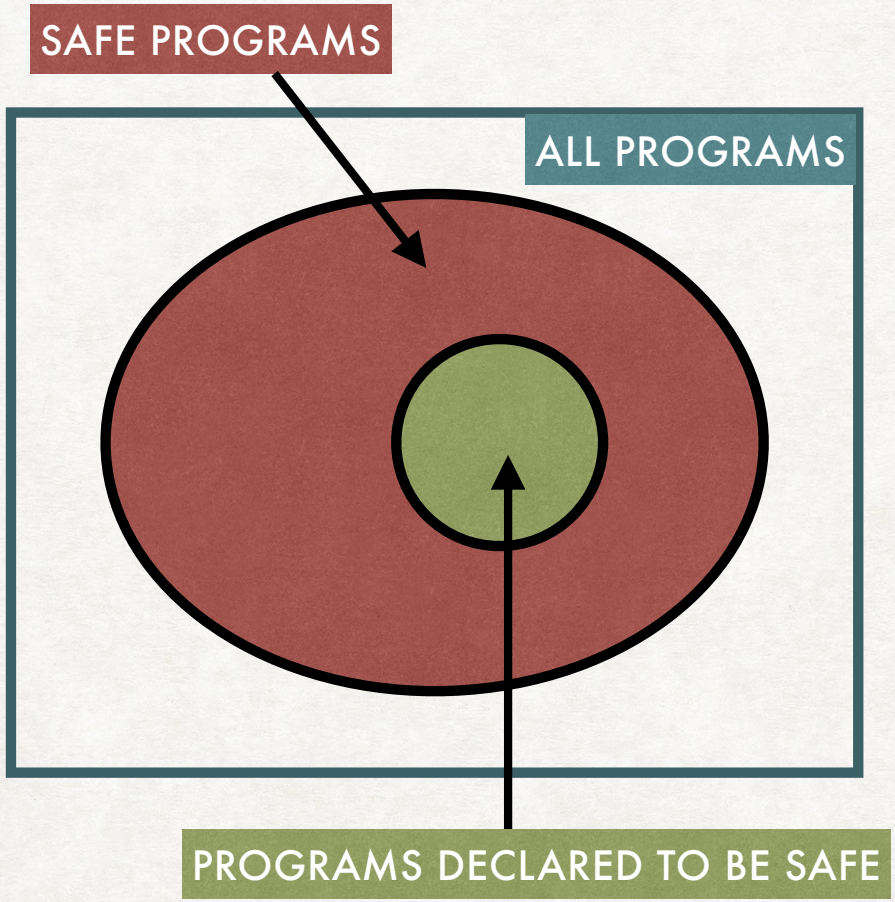
STRONGEST POST-CONDITION

- Is the Strongest Post-condition based verification procedure sound?
  - Yes, by construction.
- Is the Strongest Post-condition based verification procedure complete?
  - Relatively Complete.
  - Relies on validity (modulo theory) of an FOL formula.
  - If the FOL theory is incomplete, then there could be valid formulae in the theory, which are not valid modulo the axioms. E.g. Peano Arithmetic.
  - However, the Theory of Reals (and Theory of Integers) are both complete, and hence the procedure is also complete.

SOUNDNESS AND COMPLETENESS

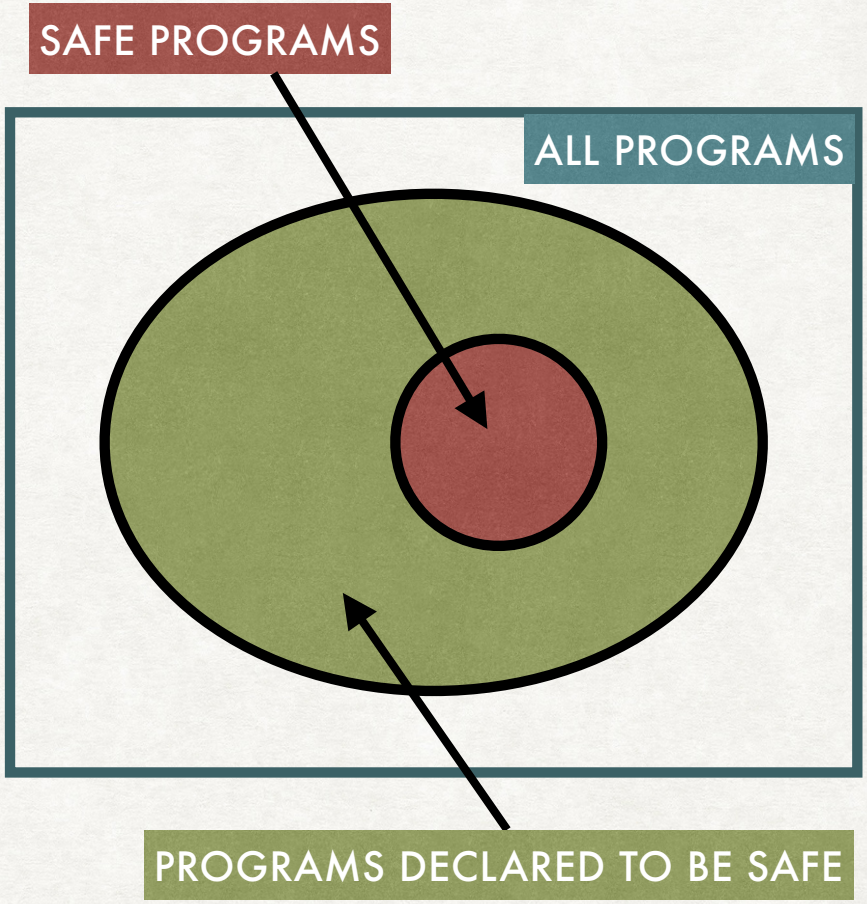
- What is an example of a trivial sound but in-complete verification procedure?
  - Simply say all programs are un-safe
- What is an example of a trivial un-sound but complete verification procedure?
  - Simply say all programs are safe

SOUNDNESS AND COMPLETENESS



Sound, In-complete
Verification Procedure

Examples : Strongest Post-condition, Weakest Pre-condition, Hoare Logic…



Un-sound, complete
Verification Procedure

Examples : Testing, Fuzzing, Bounded Model-Checking…

ISSUES WITH SP

- Loops can lead to unbounded computation.
- sp computation for assignments is tedious and inefficient. Requires quantifier elimination after every assignment statement.
  - Ideally, we want a syntactic, linear-time method for computing the FOL formula.