STRONGEST POST-CONDITION

SYMBOLIC EXECUTION IN THE FORWARD DIRECTION

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

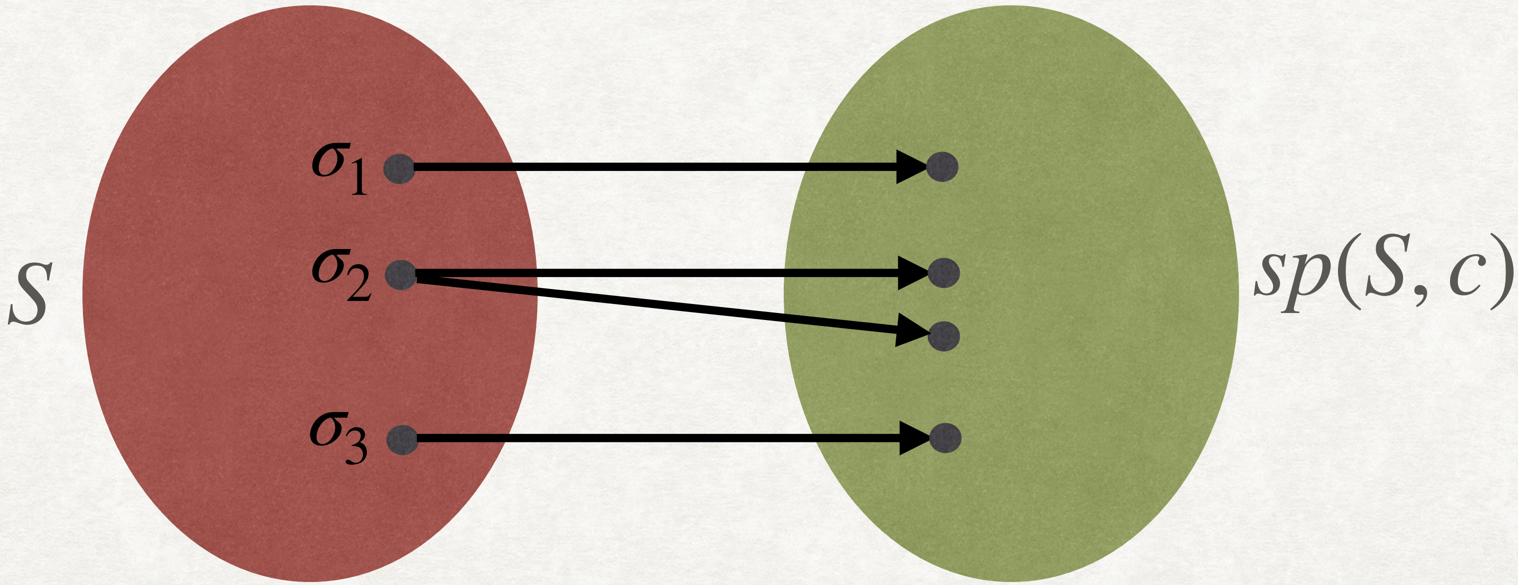
$$sp(S, c) \triangleq \{\sigma' \mid \exists \sigma \in S\,.\ (\sigma, c) \hookrightarrow^* (\sigma', \mathsf{skip})\}$$

STRONGEST POST-CONDITION

SYMBOLIC EXECUTION IN THE FORWARD DIRECTION

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

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

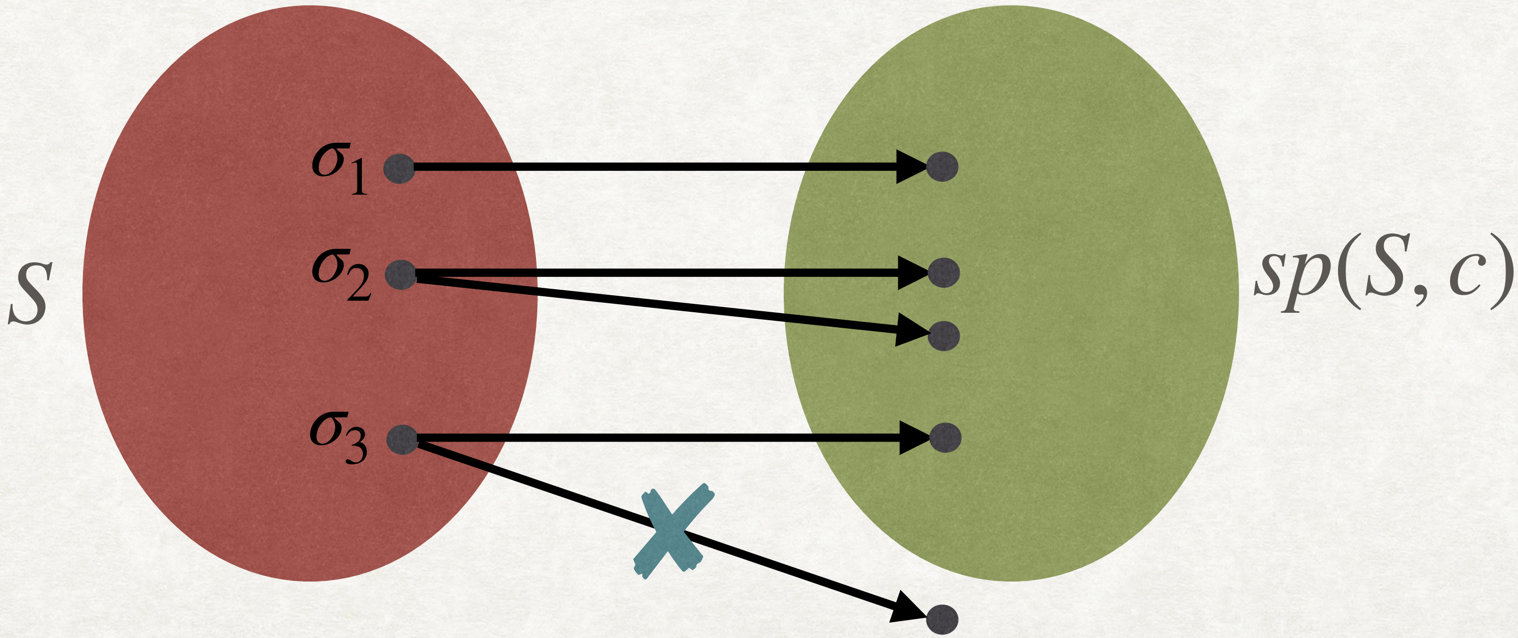


STRONGEST POST-CONDITION

SYMBOLIC EXECUTION IN THE FORWARD DIRECTION

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

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

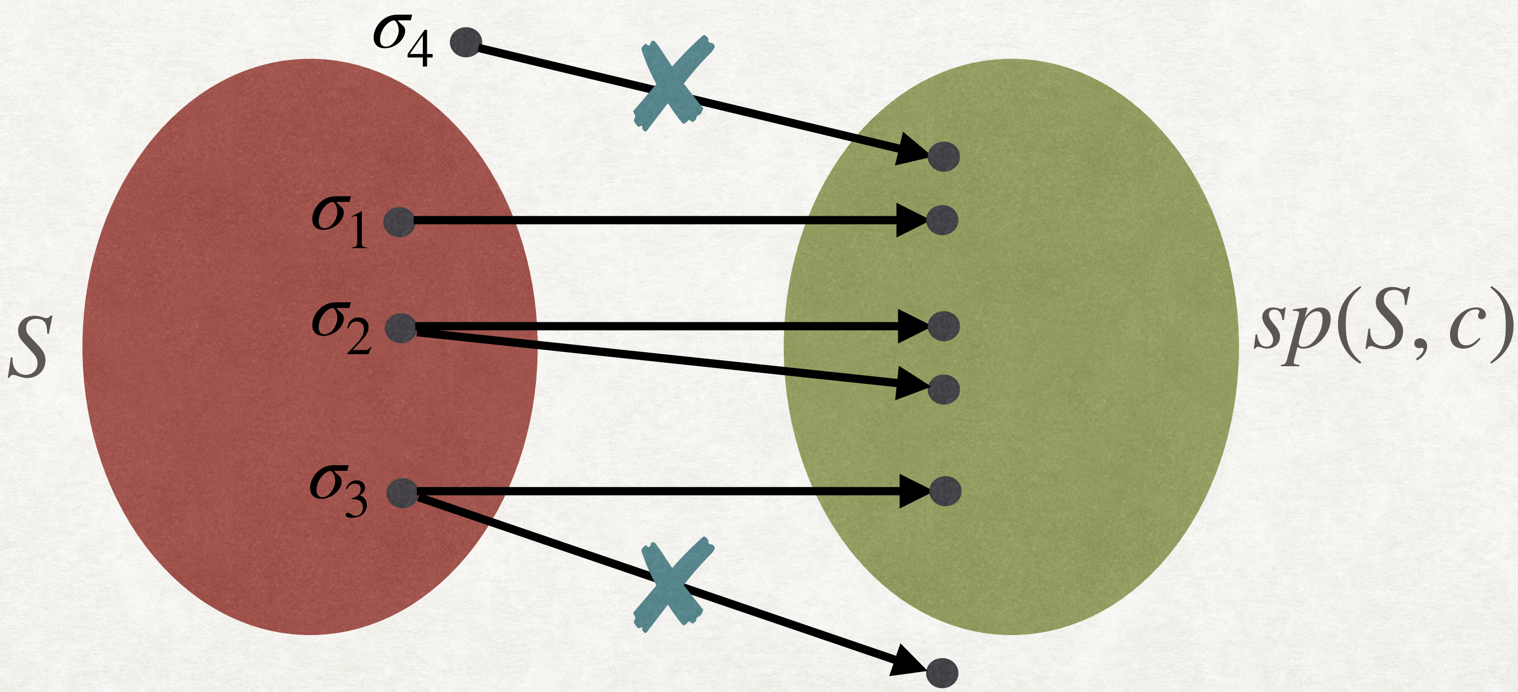


STRONGEST POST-CONDITION

SYMBOLIC EXECUTION IN THE FORWARD DIRECTION

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

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

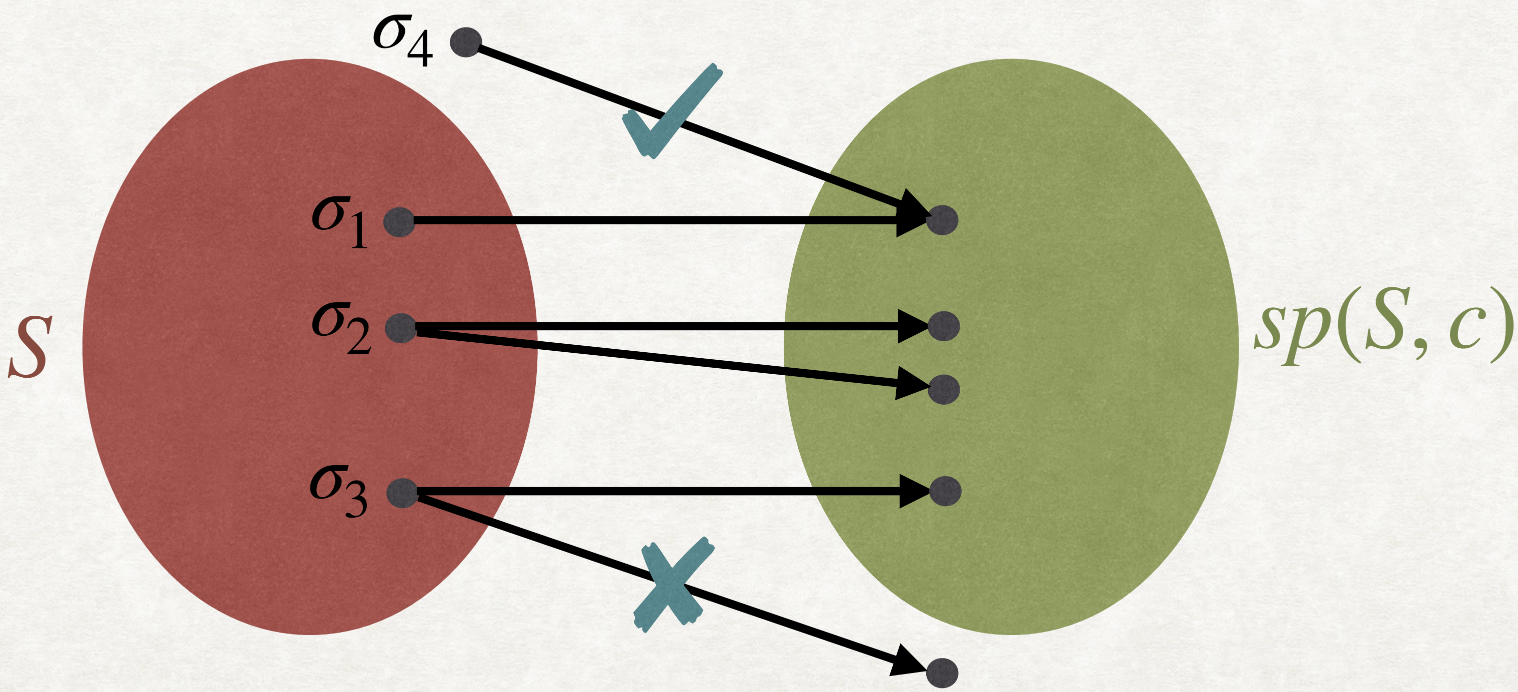


STRONGEST POST-CONDITION

SYMBOLIC EXECUTION IN THE FORWARD DIRECTION

- Given a set of states S and a command c, the strongest post-condition operator $sp(S, c)$ 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 operator $sp(S, c)$ 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:

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

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

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

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

STRONGEST POST-CONDITION

EXAMPLE

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

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

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

Eliminate x and y

Substitute x’ and y’ with x and y

STRONGEST POST-CONDITION

EXAMPLE

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

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

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

Alternative Formulation for Assignment Statement:

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

STRONGEST POST-CONDITION

MORE EXAMPLES

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

STRONGEST POST-CONDITION

MORE EXAMPLES

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

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

STRONGEST POST-CONDITION

MORE EXAMPLES

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

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

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

STRONGEST POST-CONDITION

MORE EXAMPLES

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

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

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

STRONGEST POST-CONDITION

MORE EXAMPLES

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

$\triangleq \text{y} > 0$

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

STRONGEST POST-CONDITION

MORE EXAMPLES

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

$\triangleq \mathsf{y} > 0$

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

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

STRONGEST POST-CONDITION

MORE EXAMPLES

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

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

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

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

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

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

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

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

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

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

STRONGEST POST-CONDITION

EXAMPLES

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

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

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

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

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

EXAMPLES

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

STRONGEST POST-CONDITION

COMPOUND STATEMENTS

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

STRONGEST POST-CONDITION

COMPOUND STATEMENTS

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

STRONGEST POST-CONDITION

COMPOUND STATEMENTS

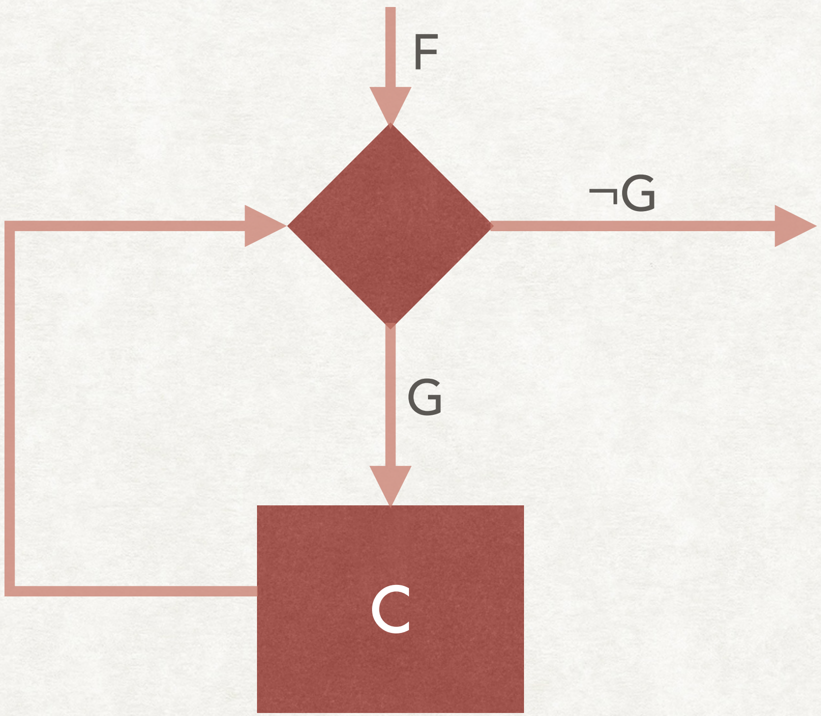
- $sp(\mathrm{F}, \mathrm{c;c'}) \triangleq sp(sp(\mathrm{F}, \mathrm{c}), \mathrm{c'})$
- $sp(\mathrm{F}, \mathrm{if(G)\ then\ c\ else\ c'}) \triangleq sp(\mathrm{F} \wedge \mathrm{G}, \mathrm{c}) \vee sp(\mathrm{F} \wedge \neg\mathrm{G}, \mathrm{c'})$

HOMEWORK: PROVE USING DEFINITION OF SP

STRONGEST POST-CONDITION

WHILE LOOPS

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

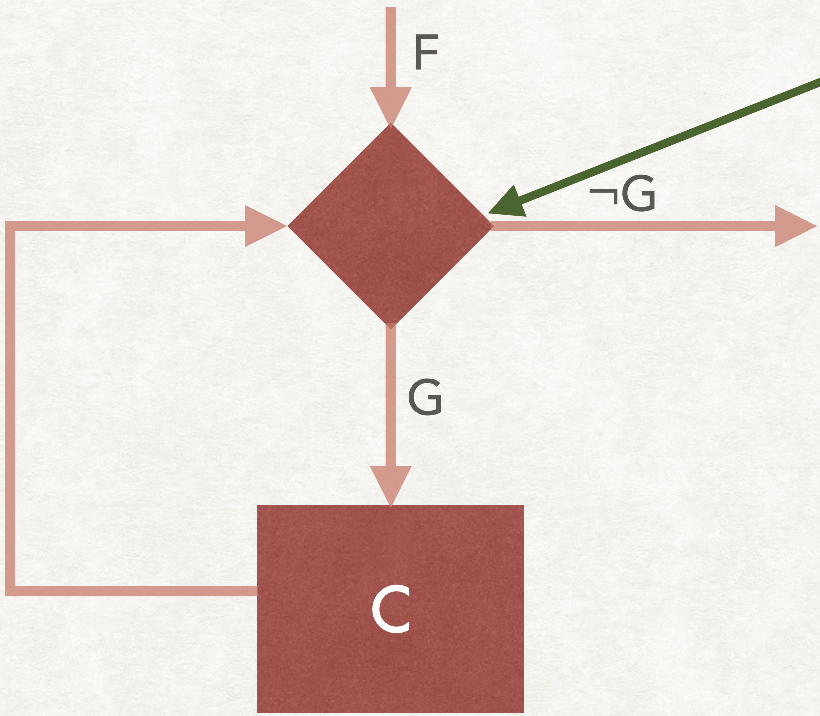


STRONGEST POST-CONDITION

WHILE LOOPS

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

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

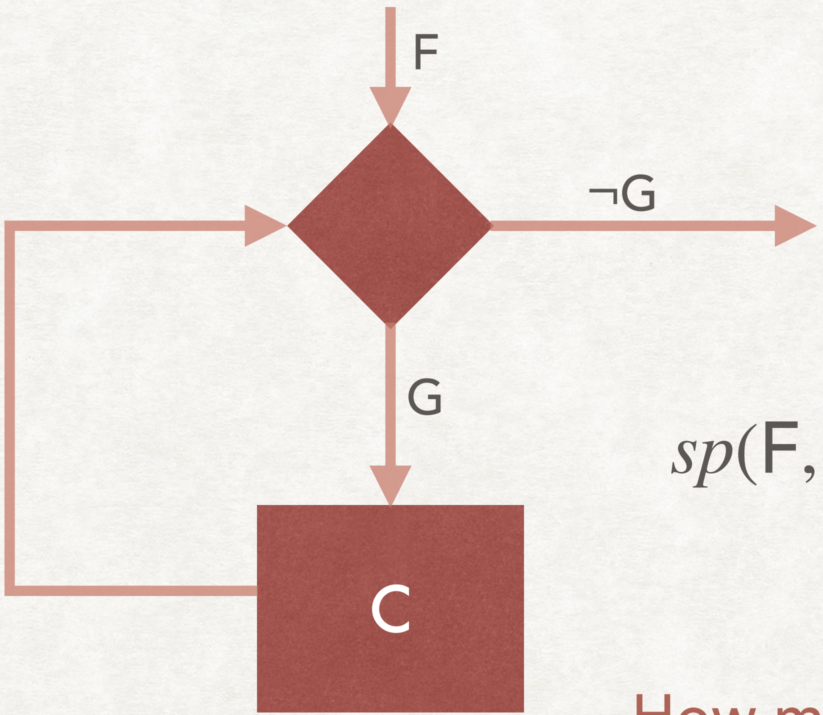


Iteration i	States possible upto iteration i
0	F
1	$F \lor sp(F \land G, c)$
2	$F \lor sp(F \land G, c) \lor sp(sp(F \land G, c) \land G, c)$
…	…

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?

Until $F^k \rightarrow F^{k-1}$

WHILE LOOPS

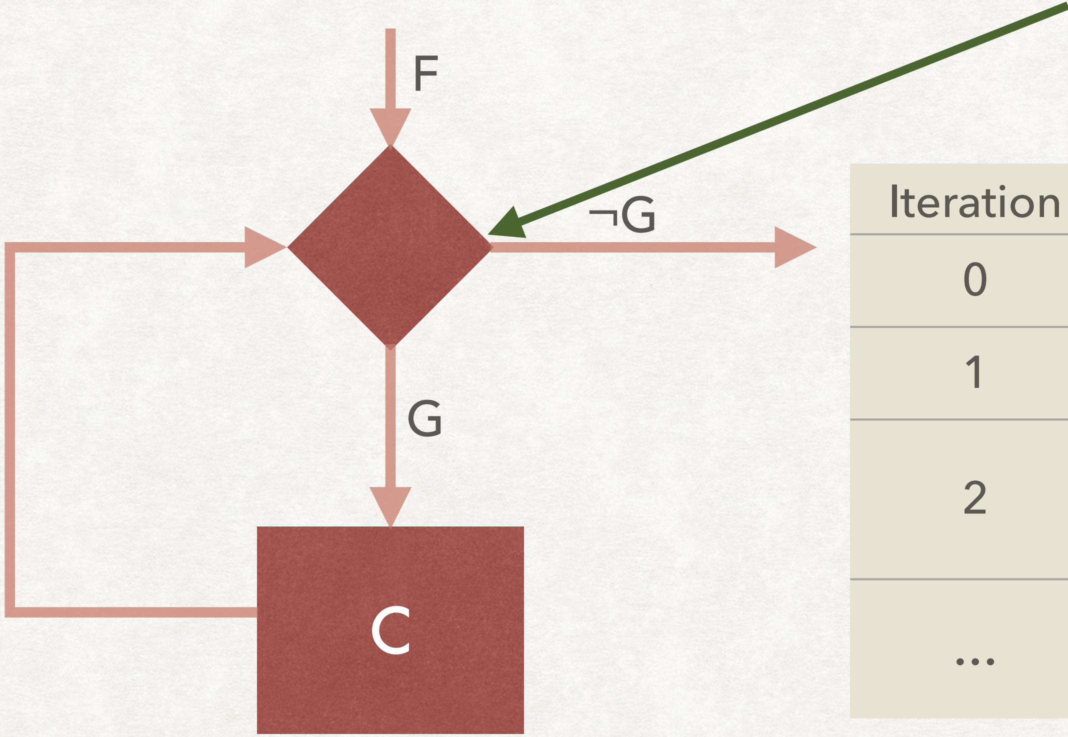
EXAMPLES

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

WHILE LOOPS

EXAMPLES

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

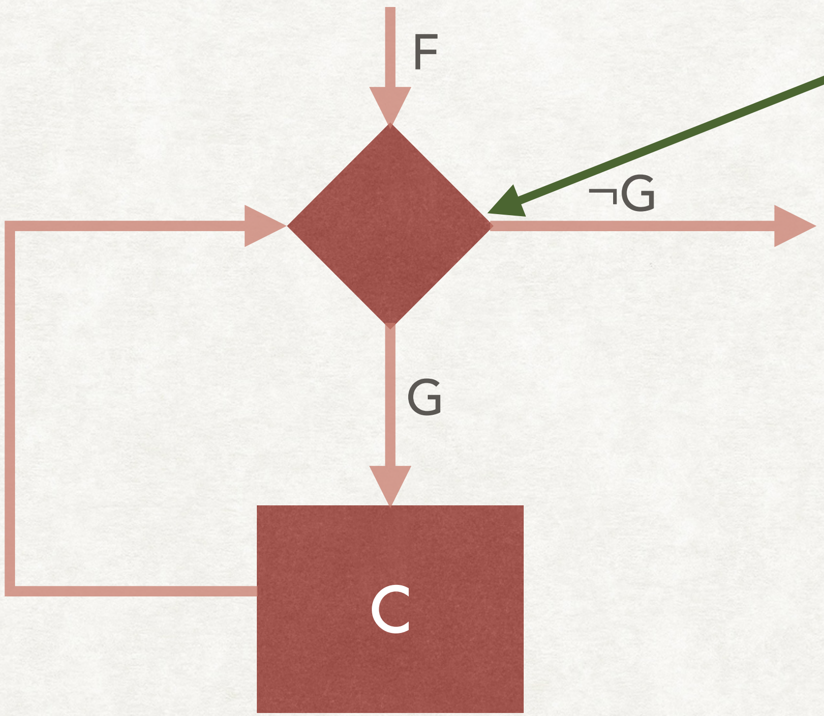


Iteration i	States possible upto iteration i
0	F
1	F $\vee$ sp(F $\wedge$ G, c)
2	F $\vee$ sp(F $\wedge$ G, c) $\vee$ sp(sp(F $\wedge$ G, c) $\wedge$ G, c)
…	…

WHILE LOOPS

EXAMPLES

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



Iteration i	States possible upto iteration i
0	$x < 0$
1	$x < 1$
2	$x < 2$
…	…
11	$x < 11$

WHILE LOOPS

EXAMPLES

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

WHILE LOOPS

EXAMPLES

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

WHILE LOOPS

EXAMPLES

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

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.

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);
```

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

$$\begin{aligned} sp(error = 0 \wedge i = 0 \wedge n \geq 0 \wedge i < n, 3:) &\equiv \exists i, n, error\,.\, error = 0 \wedge i = 0 \wedge \\ &\quad n \geq 0 \wedge i < n \wedge i' = i + 1 \wedge frame(i) \\ &\equiv error = 0 \wedge i = 1 \wedge n > 0 \end{aligned}$$

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

$$\vdots$$

COMPLETE EXAMPLE

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

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

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

COMPLETE EXAMPLE

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

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

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

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

SOUNDNESS AND COMPLETENESS

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