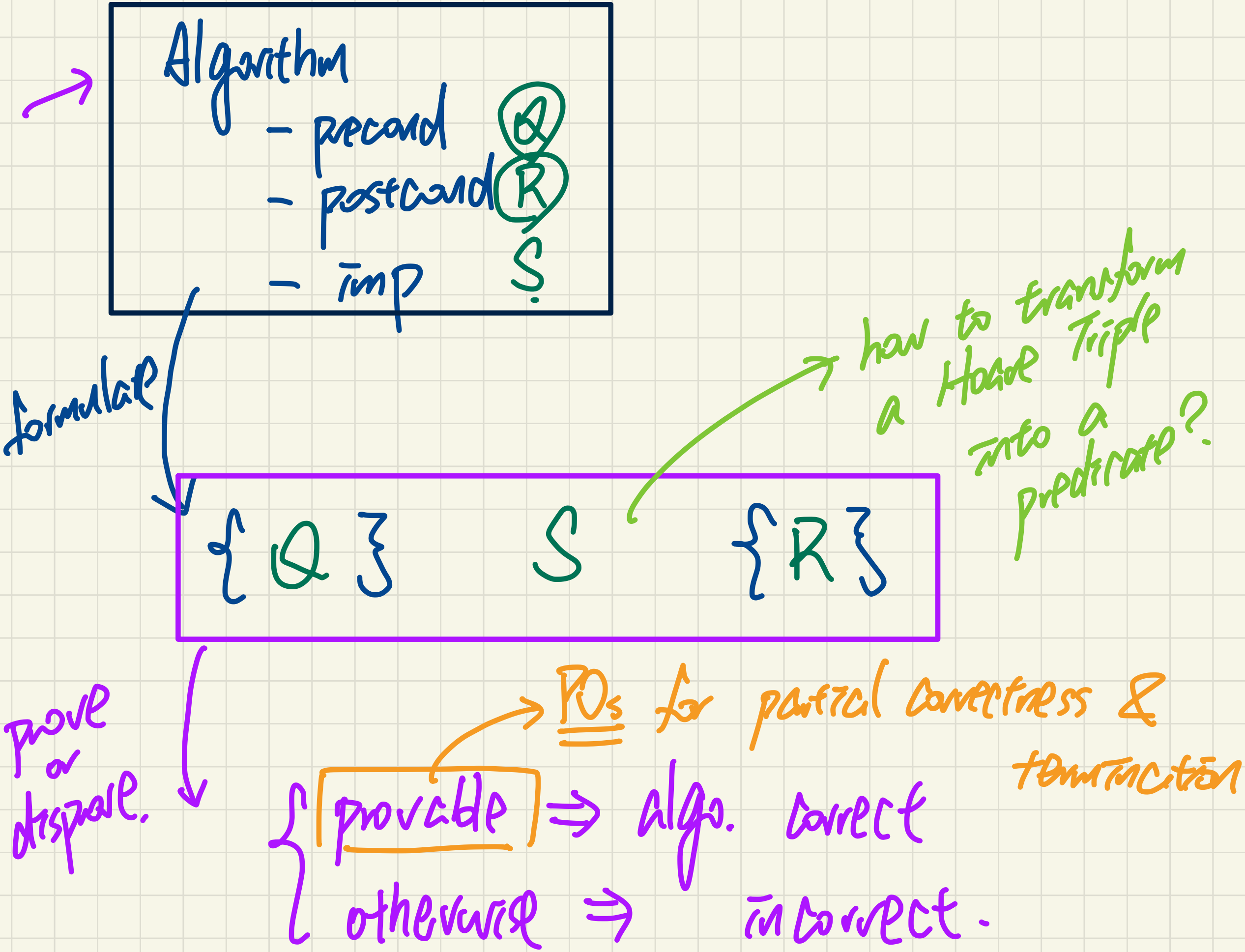
Lecture 16 - March 30

Program Verification

Weakest Precondition (WP)
WP Rules

Announcements

- Lab3 due tomorrow → Friday, noon
- ProgTest2 → level of difficulty ≈ EECS1021/1022



Algorithm
- precond Q
- postcond R
- imp S

formulate

$\{Q\}\ S\ \{R\}$

how to transform a Hoare Triple into a predicate?

prove or disprove.

POs for partial correctness & termination

- provable ⇒ algo. correct
- otherwise ⇒ incorrect.

Hoare Triple as a Predicate

$$\{Q\}\ S\ \{R\} \equiv Q \Rightarrow wp(S, R)$$

→ if Q is false, program is trivially correct.

express: Q is stronger than wp(S, R)

⇒ fails to hold if Q is weaker than wp(S, R)

Hoare Triple

Correct Program

precond.

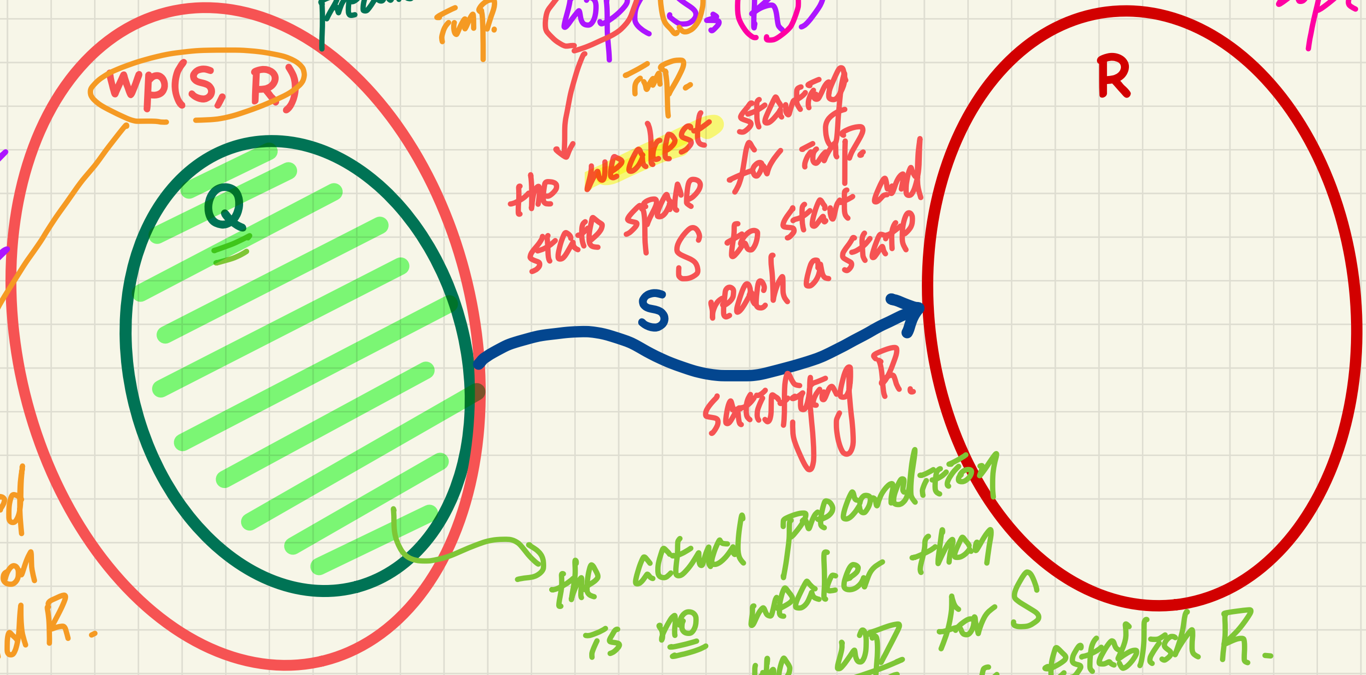
imp.

wp(S, R)

imp.

postcond.

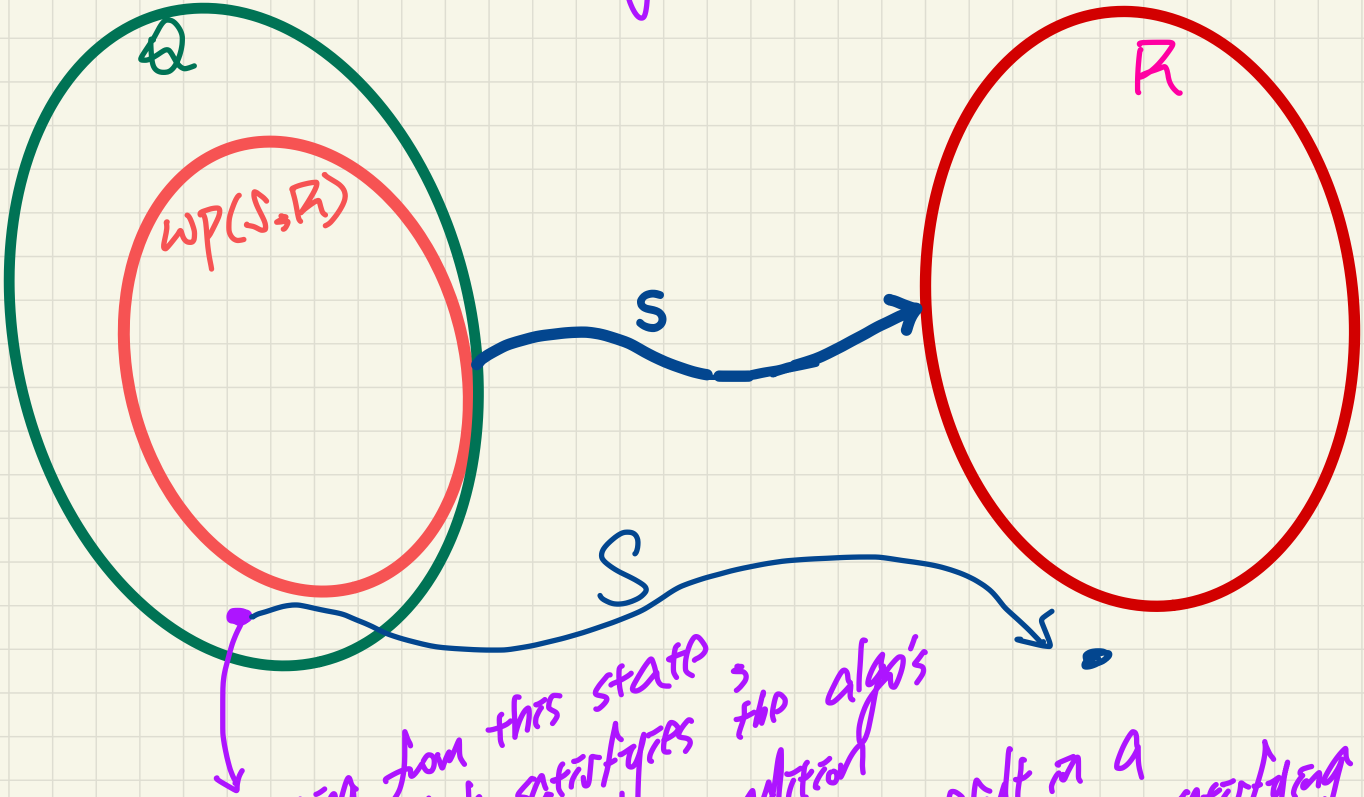
the weakest starting state space for imp. S to start and reach a state satisfying R.



calculated based on S and R.

the actual precondition is no weaker than the wp for S to establish R.

Incorrect Program



starting from this state which satisfies the algo's precondition will not result in a state satisfying R pre-state value of b (value of b at the beginning of algorithm)

$$\{b_0 > a\}\ b := b - a\ \{\boxed{b = b_0 - a}\}$$

usually, subscript 0 is omitted

tmp?

the post-state value of b equals the pre-state value of b minus a

3342:

b → pre-state

b' → post-state

4315

b_0 → pre-state

b → post-state

Lecture

Program Verification

Rules of wp Calculus

Rules of Weakest Precondition: Assignment

→ base case for wp calculation

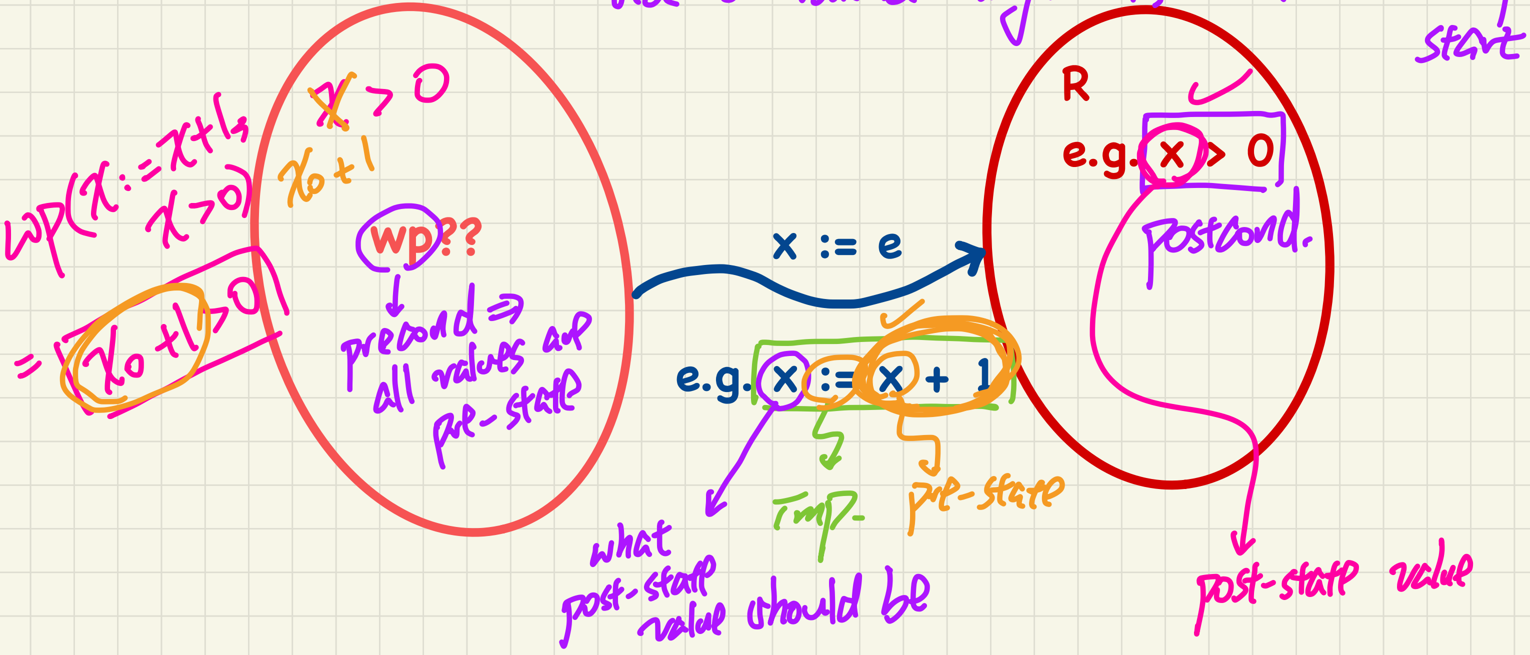
$\{Q\}\ x := e\ \{R\}$

$\hookrightarrow Q \Rightarrow \boxed{wp(x := e, R)}$

$R[x := e]$

$$wp(\overset{S}{\boxed{x := e}}, R) = R[x := e]$$

↓ to achieve the postcond. R, via a variable assignment, what's the wp to start?



$wp(x := x+1, x > 0)$

$= (x_0 + 1) > 0$

$\cancel{x} > 0$

$x_0 + 1$

wp??

↓ precond ⇒ all values at pre-state

x := e

e.g. x := x + 1

what post-state value should be

Tmp.

pre-state

R

e.g. x > 0

postcond.

post-state value

$$\boxed{wp}(x := 23,\ \underline{\underline{x = 46}})$$

$$= \{\text{ wp value for } := \}$$

$$\underline{\underline{x}} = 46\,[x := 23]$$

$$= 23 = 46$$

$$= \boxed{False}$$

acceptable input values: no input values can guarantee that "x := 23" will establish 46.

In case, you're better off just fixing S or R.

the only way to have a correct program is:

$$\{\underset{}{\overset{Q}{\underline{False}}}\}$$

$$x := 23$$

$$\{x = 46\}$$

$\because False \Rightarrow False \equiv T$

Correctness of Programs: Assignment (1)

What is the weakest precondition for a program `x := x + 1` to establish the postcondition $x > x_0$?

$$\{??\}\ \texttt{x := x + 1}\ \{x > x_0\}$$

$$wp(x := \overset{e}{\boxed{x+1}},\ \underline{x > x_0})$$

$$= \{\text{wp rule of } :=\}$$

$$\underline{x > x_0}\,[x := x_0 + 1]$$

$$= x_0 + 1 > x_0$$

$$= 1 > 0 = \boxed{\text{True}}$$

any precondition will be correct

$\Rightarrow$ True

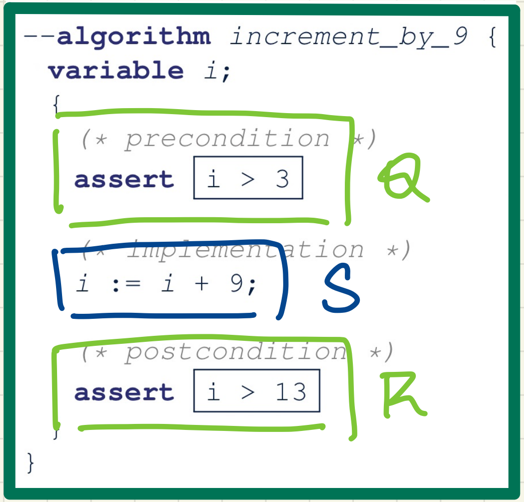
Correctness of Programs: Assignment (2)

Exercise .

What is the weakest precondition for a program `x := x + 1` to establish the postcondition $x > x_0$?

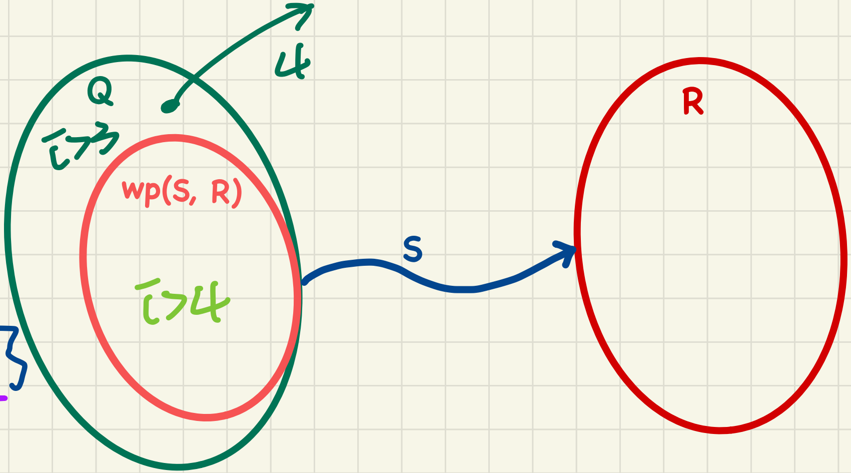
$$\{??\}\ \texttt{x := x + 1}\ \{x = 23\}$$

Program Correctness: Revisiting Example (1)



```
--algorithm increment_by_9 {
  variable i;
  {
    (* precondition *)
    assert i > 3
    (* implementation *)
    i := i + 9;
    (* postcondition *)
    assert i > 13
  }
}
```

$$\{Q\}\ S\ \{R\} \equiv Q \Rightarrow wp(S, R)$$

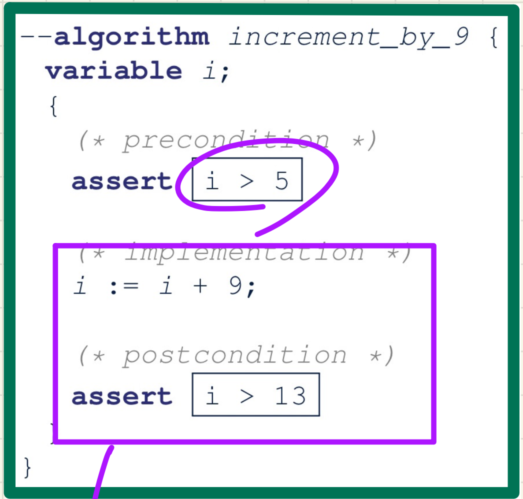


$$\{i > 3\}\ i := i + 9\ \{i > 13\}$$

$$\Leftrightarrow$$

$$i > 3 \Rightarrow \overset{i>4}{\boxed{wp(i := i+9,\ i > 13)}}$$

Program Correctness: Revisiting Example (2)



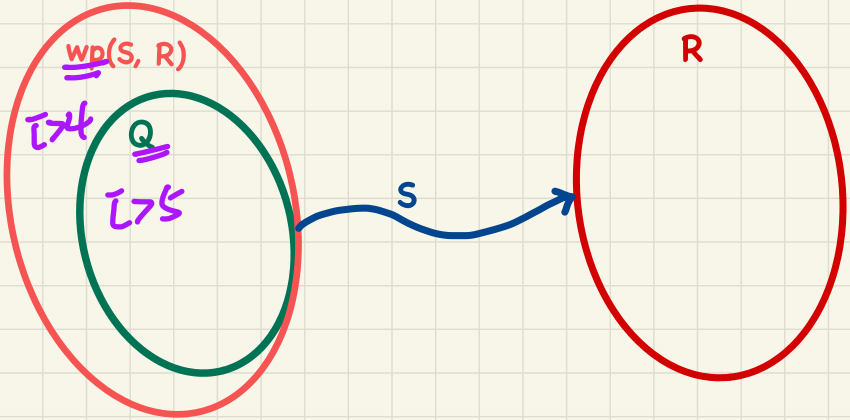
```
--algorithm increment_by_9 {
  variable i;
  {
    (* precondition *)
    assert i > 5

    (* implementation *)
    i := i + 9;

    (* postcondition *)
    assert i > 13
  }
}
```

$$\{Q\}\ S\ \{R\} \equiv Q \Rightarrow wp(S, R)$$

$wp(i := i+9, i > 13)$

$= i > 4$

Argue: $i > 5 \Rightarrow i > 4$



Rules of Weakest Precondition: Conditionals

branch 1 imp.

branch 2 imp.

$$wp(\text{if } B \text{ then } S1 \text{ else } S2 \text{ end}, R)$$

✓

$$B \Rightarrow wp(S_1, R)$$

$$\land$$

$$\neg B \Rightarrow wp(S_2, R)$$

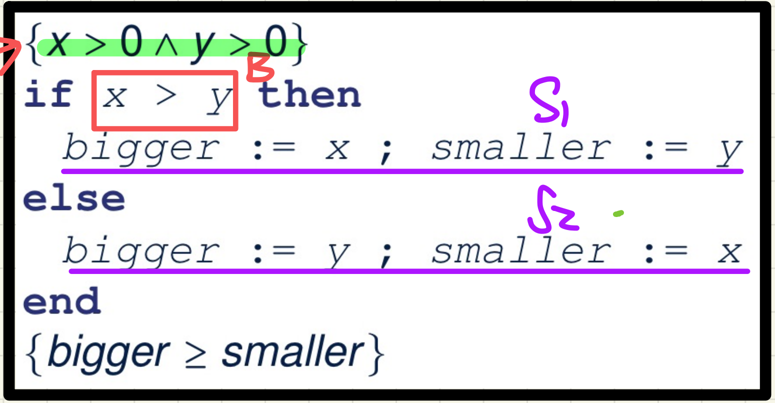
both branches should be able to establish the R. by the corresponding statement.

$$\{Q\}\ \overbrace{\text{if } B \text{ then } S_1 \text{ else } S_2}^{S}\ \{R\}$$

$$Q \Rightarrow \underbrace{\left(\begin{array}{l} B \Rightarrow wp(S_1, R) \\ \land \\ \neg B \Rightarrow wp(S_2, R) \end{array}\right)}_{wp(S,\ R)}$$

Correctness of Programs: Conditionals

Is this program correct?



$\{x > 0 \wedge y > 0\}$

```
if x > y then
  bigger := x ; smaller := y
else
  bigger := y ; smaller := x
end
```

$\{bigger \geq smaller\}$

B

S_1

S_2

(Step 3)

Argue: $x > 0 \wedge y > 0 \overset{?}{\Rightarrow} wp$

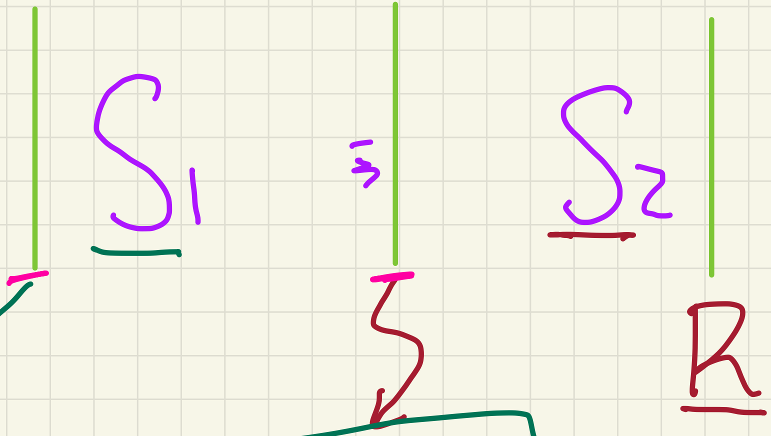
(Step 1) Formulate Hoare Triple

$\{x > 0 \wedge y > 0\}$ $\boxed{\text{if } B \text{ then } S_1 \text{ else } S_2}$ $\{bigger \geq smaller\}$

(Step 2) Calculate wp(if B then S_1 else S_2 , $bigger \geq smaller$).

Exercise.

$$wp(\underbrace{\check{S_1}}_{\text{phase 1}} ; \underbrace{\check{S_2} , R}_{\text{remaining phases}})$$



$$wp(S_1 ; wp(S_2 , R)) \qquad \boxed{wp(S_2 , R)}$$

Correctness of Programs: Sequential Composition

Is $\{$ ***True*** $\}$ `tmp := x; x := y; y := tmp` $\{ x > y \}$ correct?

(Step 1) Calculate $wp(\boxed{tmp := x}; \boxed{x := y; y := tmp}, x > y)$

$= \{ wp \text{ rule for } ; \}$ ✓

$wp(tmp := x, wp(\boxed{x := y}; \boxed{y := tmp}, x > y))$

$= wp(tmp := x, wp(x := y, wp(y := tmp, x > y)))$

$= \{ wp \text{ rule of } := \}$

$wp(tmp := x, wp(x := y, x > tmp))$

$= \{ wp \text{ rule of } := \}$

$wp(tmp := x, y > tmp)$

$= \{ \overset{wp}{\text{rule}} \text{ of } := \}$

$\boxed{y > x}$

wp-

(Step 2) $True \Rightarrow y > x \equiv \boxed{y > x}$

~~=~~ not a theorem.
counter ex:
$x = 2$
$y = 1$