Correctness Triples

- A **correctness triple** (a.k.a. "Hoare triple," after C.A.R. Hoare; or usually simplified to **"triple"**), written as $\{p\}\, S\, \{q\}$ is a program S plus its specification predicates p and q.
  - The **precondition** p (not "$\{p\}$") describes the collection of states that we want to execute S in.
  - The **postcondition** q (not "$\{q\}$") describes the collection of states we expect S terminates in.
  - *Informally*, a triple $\{p\}\, S\, \{q\}$ means "if program S runs in a state that satisfies p, then we expect the execution of S terminates in some state (or states) satisfies q".

  Here are some examples of correctness triples:
  - $\{x \le 2\}\, x := x + 3\, \{x < 6\}$
  - $\{x \ge 0\}\, S\, \{y^2 \le x < (y+1)^2\}$

A tripe can "make no sense": the execution of S in a state satisfying p can never ends in some state satisfying q. So here, let us understand the satisfaction and validity of a triple.

(Satisfaction and Validity under Total Correctness)

- The triple $\{p\}\, S\, \{q\}$ is **totally correct in** σ (or σ **satisfies the triple under total correctness**), written as $\sigma \vDash_{tot} \{p\}\, S\, \{q\}$, if and only if it is the case that "**if** σ satisfies p, **then** the execution of S in σ always terminates (without error) in states satisfying q".
  - In other words, $\sigma \vDash_{tot} \{p\}\, S\, \{q\} \Leftrightarrow (\sigma \ne \bot) \wedge \big((\sigma \vDash p) \to (M(S,\sigma) \vDash q)\big)$.

Without specification, while we analyze whether state $\boldsymbol{\sigma}$ satisfies triple $\{\boldsymbol{p}\}\, \boldsymbol{S}\, \{\boldsymbol{q}\}$, we always assume that $\boldsymbol{\sigma} \ne \bot$.

1. True or False.
   a. $\{x = -5\} \vDash_{tot} \{x > 0\}\, x := x + 1\, \{x > 0\}$
   True. Since $\{x = -5\}$ doesn't satisfy the precondition $x > 0$, so the triple satisfied.

   b. $\{y = 1\} \vDash_{tot} \{x > 0\}\, x := x + 1\, \{x > 0\}$
   True. Since $\{y = 1\}$ is not proper for the precondition $x > 0$ so it cannot satisfy the precondition, so the triple satisfied.

   c. $\{x = -1\} \vDash_{tot} \{x \le 0\}\, x := x + 1\, \{x \ge 0\}$
   True. Since $\{x = -1\}$ satisfies the precondition $x \le 0$, so we need to execute $x := x + 1$, and $M(x := x + 1\,,\ \{x = -1\}) = \big\{\{x = 0\}\big\}$, and it satisfies the postcondition $x \ge 0$.

   d. $\{x = -5\} \vDash_{tot} \{x \le 0\}\, x := x + 1\, \{x \ge 0\}$
   False. Since $\{x = -5\}$ satisfies the precondition $x \le 0$, so we need to execute $x := x + 1$, and $M(x := x + 1\,,\ \{x = -5\}) = \big\{\{x = -4\}\big\}$, and it doesn't satisfy the postcondition $x \ge 0$.

   e. $\{x = 0\} \vDash_{tot} \{x \le 0\}\, x := 1/x\, \{x \ge 0\}$
   False. Since $\{x = 0\}$ satisfies the precondition $x \le 0$, so we need to execute $x := 1/x$, and $M(x := 1/x\,,\ \{x = 0\}) = \{\bot_e\}$, and it doesn't satisfy the postcondition $x \ge 0$.

- From the above examples, we can see that "$\sigma \vDash_{tot} \{p\}\, S\, \{q\}$" might not give us much information about executing S in σ. But on the other hand, "$\sigma \nvDash_{tot} \{p\}\, S\, \{q\}$" shows that $\sigma \vDash p$ and the execution of S in σ doesn't end in states satisfying q.

- The triple $\{p\}\, S\, \{q\}$ is **totally correct** (or the triple is **valid under total correctness**) if and only if $\sigma \vDash_{tot} \{p\}\, S\, \{q\}$ for all $\sigma \in \Sigma$ (Recall that Σ is the set of well-formed states). We write $\vDash_{tot} \{p\}\, S\, \{q\}$.
  - $\vDash_{tot} \{p\}\, S\, \{q\}$ means $\forall \sigma. \sigma \vDash_{tot} \{p\}\, S\, \{q\}$.
  - $\nvDash_{tot} \{p\}\, S\, \{q\}$ means the triple is invalid: $\exists \sigma. \sigma \nvDash_{tot} \{p\}\, S\, \{q\}$.

2. True of False
   a. $\vDash_{tot} \{x > 0\}\, x := x + 1\, \{x > 0\}$ True
      $\vDash_{tot} \{x > 0\}\, x := x - 1\, \{x > 0\}$ False, we can find $\{x = 1\} \nvDash \{x > 0\}\, x := x - 1\, \{x > 0\}$

(Satisfaction and Validity under Partial Correctness)

- The triple $\{p\}\, S\, \{q\}$ is **partially correct in** σ (or σ **satisfies the triple under partial correctness**), written as $\sigma \vDash \{p\}\, S\, \{q\}$, if and only if it is the case that "**if** σ satisfies p, **then if** the execution of S in σ can terminate without an error, it terminates in states satisfying q".
  - In other words, $\sigma \vDash \{p\}\, S\, \{q\} \Leftrightarrow (\sigma \neq \bot) \wedge \big((\sigma \vDash p) \rightarrow \forall \tau \in M(S, \sigma). \tau \neq \bot \rightarrow \tau \vDash q\big)$;
    or equivalently, $\sigma \vDash \{p\}\, S\, \{q\} \Leftrightarrow (\sigma \neq \bot) \wedge \big((\sigma \vDash p) \rightarrow M(S, \sigma) - \bot \vDash q\big)$.

- The triple $\{p\}\, S\, \{q\}$ is **partially correct** (or the triple is **valid under partial correctness**) if and only if $\sigma \vDash \{p\}\, S\, \{q\}$ for all $\sigma \in \Sigma$. We write $\vDash \{p\}\, S\, \{q\}$.

3. True or False.
   a. $\{x = -5\} \vDash \{x > 0\}\, x := x + 1\, \{x > 0\}$ True.
   b. $\{x = -1\} \vDash \{x \le 0\}\, x := x + 1\, \{x \ge 0\}$ True.
   c. $\{x = -5\} \vDash \{x \le 0\}\, x := x + 1\, \{x \ge 0\}$ False.
   d. $\{x = 0\} \vDash \{x \le 0\}\, x := 1/x\, \{x \ge 0\}$
      True. Since $\{x = 0\}$ satisfies the precondition $x \le 0$, so we need to execute $x := 1/x$, and $M(x := 1/x\, ,\ \{x = 0\}) - \bot = \emptyset$, and it satisfies the postcondition $x \ge 0$.

4. If $\sigma \vDash p$ and $M(S, \sigma) = \{\bot\}$, then:
   a. Does $\sigma \vDash_{tot} \{p\}\, S\, \{q\}$? No
   b. Does $\sigma \vDash \{p\}\, S\, \{q\}$? Yes

- The difference between two correctness is whether we accept that executing S in σ ends with $\bot$. We can say: $\sigma \vDash_{tot} \{p\}\, S\, \{q\} \Leftrightarrow\ (\sigma \vDash \{p\}\, S\, \{q\}) \wedge \bot \notin M(S, \sigma)$.

5. True or False:
   a. $\vDash_{tot} \{F\}\, S\, \{q\}$ True, nothing can satisfy the precondition.
   b. $\vDash_{tot} \{p\}\, S\, \{T\}$ False, it is not true for some $\sigma \vDash p$ such that $\bot \in M(S, \sigma)$
   c. $\vDash \{F\}\, S\, \{q\}$ True, nothing can satisfy the precondition.
   d. $\vDash \{p\}\, S\, \{T\}$ True, for any state $\sigma \vDash p$, $\forall \tau \in M(S, \sigma). \tau = \bot \vee \tau \vDash T$

6. Let $W \equiv \textbf{while}\ k \neq 0\ \textbf{do}\ k := k - 1\ \textbf{od}$. Decide true or false.
   a. $\vDash_{tot} \{k \ge 0\}\, W\, \{k = 0\}$ True.
   b. $\vDash_{tot} \{k = -1\}\, W\, \{k = 0\}$ False. W will diverge in a state with $k = -1$.

c. $\models \{k = -1\}\ W\ \{k = 0\}$ True.

d. $\models \{T\}\ W\ \{k = 0\}$ True. If $k < 0$ then W diverges or else W ends with $k = 0$.

e. $\models_{tot} \{T\}\ W\ \{k = 0\}$ False.

7. Finish the following semantic equalities (remind that, we assume that $\sigma \neq \perp$).

a.
$$\begin{aligned} \sigma \models_{tot} \{p\}\ S\ \{q\} &\Leftrightarrow (\sigma \models p) \to (M(S,\sigma) \models q) \\ &\Leftrightarrow (\sigma \not\models p) \vee (M(S,\sigma) \models q) \\ &\Leftrightarrow (\sigma \not\models p) \vee \forall \tau \in M(S,\sigma).\ \tau \models q \end{aligned}$$

b.
$$\begin{aligned} \sigma \models \{p\}\ S\ \{q\} &\Leftrightarrow (\sigma \models p) \to (M(S,\sigma) - \perp \models q) \\ &\Leftrightarrow (\sigma \not\models p) \vee (M(S,\sigma) - \perp \models q) \\ &\Leftrightarrow (\sigma \not\models p) \vee \forall \tau \in M(S,\sigma).\ \tau = \perp \vee \tau \models q \end{aligned}$$

c.
$$\begin{aligned} \sigma \not\models_{tot} \{p\}\ S\ \{q\} &\Leftrightarrow (\sigma \models p) \wedge (M(S,\sigma) \not\models q) \\ &\Leftrightarrow (\sigma \models p) \wedge \exists \tau \in M(S,\sigma).\ \tau = \perp \vee \tau \models \neg q \end{aligned}$$

d.
$$\begin{aligned} \sigma \not\models \{p\}\ S\ \{q\} &\Leftrightarrow (\sigma \models p) \wedge (M(S,\sigma) - \perp \not\models q) \\ &\Leftrightarrow (\sigma \models p) \wedge \exists \tau \in M(S,\sigma).\ \tau \neq \perp \wedge \tau \not\models q \end{aligned}$$

(Creating Valid Triples)

- When we have some valid triple(s) given to us, can we use them to create more valid triple(s)? **The validity here can be under either correctness.**

8. If we are given valid two triples, can we join them?

a. We have valid triples $\{x = k\}\ S_1\ \{x = m\}$, and $\{x = m\}\ S_2\ \{x = n\}$, what can be a postcondition for $\{x = k\}\ S_1; S_2\ \{q\}$?
It is quite easy to see that $\{x = k\}\ S_1; S_2\ \{x = n\}$ can be a valid triple.

- **[Sequence Rule]** If we have valid triples $\{p\}\ S_1\ \{q\}$ and $\{q\}\ S_2\ \{r\}$, then we have valid triple $\{p\}\ S_1; S_2\ \{r\}$.

b. What if we have triples $\{x = k\}\ S_1\ \{x \geq m\}$ and $\{x \geq m - 1\}\ S_2\ \{x = n\}$, can we still combine these two triples into $\{x = k\}\ S_1; S_2\ \{x = n\}$?
Yes, since after executing S_1 we will end up some state(s) $\tau \models x \geq m$, so τ also satisfies the precondition of S_2.

- **[Extended Sequence Rule]** If we have valid triples $\{p\}\ S_1\ \{q\}$ and $\{q'\}\ S_2\ \{r\}$, and $q \Rightarrow q'$, then we have valid triple $\{p\}\ S_1; S_2\ \{r\}$.

9. Let $\{x \geq 0\}\ S\ \{y < 0\}$ be a valid triple.

a. Is $\{x \geq 5\}\ S\ \{y < 0\}$ valid?
Yes. $x \geq 5$ is a subcollection of $x \geq 0$, if S works "well" on all states satisfying $x \geq 0$ then it also works well on a state satisfying $x \geq 5$.

- **[Strengthening Precondition]** Strengthening the precondition of valid triple doesn't affect its validity.

b. Is $\{x \geq -5\}\ S\ \{y < 0\}$ valid?

We cannot decide, since we don't know anything about the execution of S in a state σ with $-5 \leq \sigma(x) < 0$. Weakening the precondition of a valid tripe can affect its validity.

c. Is $\{x \geq 0\}\, S\, \{y \leq 0\}$ valid ?

Yes. $y < 0$ is a subcollection of $y \leq 0$, If S terminates in states satisfying $y < 0$ then those states also satisfying $y \leq 0$.

- **[Weakening Postcondition]** Weakening the postcondition of valid triple doesn't affect its validity.

d. Is $\{x \geq 0\}\, S\, \{y < -5\}$ valid ?

We cannot decide, since we only know the execution of S terminate in states satisfying $y < 0$, but we don't know whether those states satisfy $y < -5$. Strengthening the postcondition of a valid tripe can affect its validity.

e. Among $\{x \geq 0\}\, S\, \{y < 0\}$, $\{x \geq 5\}\, S\, \{y < 0\}$, and $\{x \geq 0\}\, S\, \{y \leq 0\}$, which valid triple gives us the most information?

  - Compare $\{x \geq 0\}\, S\, \{y < 0\}$ and $\{x \geq 5\}\, S\, \{y < 0\}$. The previous one tells us that S can work well whenever $x \geq 0$; the later says S can work well ONLY when $x \geq 5$. The previous one contains more information.

  - Compare $\{x \geq 0\}\, S\, \{y < 0\}$ and $\{x \geq 0\}\, S\, \{y \leq 0\}$. The previous one tells us that S can provide us an outcome with $y < 0$; the later one says S can provide us a not-so-accurate outcome with $y < 0\ or\ y = 0$. The previous one contains more information.

- In general, weakening the postcondition or strengthening the prediction makes a valid triple to *lose information* and become less useful. On the other hand, weakening the prediction or strengthening the postcondition might affect the validity of a triple. Thus, it is quite important to find ***the weakest precondition*** and/or ***the strongest postcondition*** (and maintaining the validity at the same time), to create the "good" triples.