Correctness Triples (Continue)

1. 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.
      - 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 with a state $\tau \vDash x \geq m$, so τ also satisfies the precondition of S_2.

- **[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 $\{p\}\, S_1; S_2\, \{r\}$.

(Weaker and Stronger Conditions)

2. What is the strongest condition and what is the weakest condition?
   - A condition is weaker when it is less restricted. In the previous lectures, we've seen that T means "there is no restriction on any variables, and every state can satisfy it"; so $True$ is the weakest condition.
   - On the hand, F is so restricted that no state can satisfy it; so $False$ is the strongest condition.

- We learned that we could weaken the postcondition or strengthen the prediction can fix the validity of a triple; and we could have these operations in a valid triple without creating a problem. But this doesn't mean that we should always do this.

3. Let $\{x \geq 0\}\, S\, \{y = 0\}$ be a valid triple. We can get the following valid triples by strengthening the precondition and/or weakening the postconditions:
   a. $\{x = 0\}\, S\, \{y = 0\}$ # Strengthen the precondition
   b. $\{x \geq 0\}\, S\, \{y = 0 \vee y = 1\}$ # Weaken the postcondition

   Including $\{x \geq 0\}\, S\, \{y = 0\}$, which triple gives us the most information?

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

   - Compare $\{x \geq 0\}\, S\, \{y = 0\}$ and $\{x \geq 0\}\, S\, \{y = 0 \vee y = 1\}$. The previous one tells us that S can provide us an accurate outcome with $y = 0$; the later one says S can provide us a not-so-accurate outcome with $y = 1\ or\ 0$. 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, strengthening the postcondition or weakening the prediction might affect the validity of a triple. Thus, it is quite important to find ***the weakest precondition*** and ***the strongest postcondition*** (and maintaining the validity at the same time), to create the "good" triples.

Weakest Preconditions

- w is the **weakest precondition of** S **and** q (we write $w = wp\ (S, q)$ or $w \Leftrightarrow wp\ (S, q)$) if w is a precondition for S and q that can't be weakened. In other word, $\vDash_{tot} \{w\}\ S\ \{q\}$ and there is no r weaker than w such that $\vDash_{tot} \{r\}\ S\ \{q\}$.
  - In terms of collection of states: $wp\ (S, q) = \{\sigma \in \Sigma \mid M(S, \sigma) \vDash q\}$.

1. Let's consider $w = wp(x \coloneqq x + 1, x \geq 2)$.
   - Using the backward assignment rule we can see that $w \Leftrightarrow (x + 1 \geq 2) \Leftrightarrow (x \geq 1)$.
   - If we use terms of states, we can see w is the collection of all σ that makes $M(x \coloneqq x + 1, \sigma) \vDash (x \geq 2)$. This collection can contain $\{x = 1, y = 3\}, \{x = 100, z = 1, y = 4\}$ … In general, we can say that is the collection of states that satisfy $x \geq 1$.

2. Let $w = wp\ (S, q)$. Decide true or false.
   a. If $\vDash_{tot} \{r\}\ S\ \{q\}$, we have $w \Rightarrow r$.
      False. w should be the weakest precondition.
   b. If $r \Rightarrow w$, then $\vDash_{tot} \{r\}\ S\ \{q\}$. True.
      - To sum up, $\big(w = wp\ (S, q)\big) \Leftrightarrow \vDash_{tot} \{w\}\ S\ \{q\} \wedge \forall r. \vDash_{tot} \{r\}\ S\ \{q\} \leftrightarrow (r \to w)$.

   c. If $\sigma \nvDash w$, then we know nothing interesting about $M(S, \sigma)$.
      False. Since any w is the most general precondition for S and q, if a state σ doesn't satisfy it then $M(S, \sigma) \nvDash q$.
   d. If S is deterministic, then $\vDash \{\neg w\}\ S\ \{\neg q\}$.
      True. For any state σ, if $\sigma \vDash \neg w$ and $M(S, \sigma) = \{\tau\}$, we must have $\tau \nvDash q$; in other words, $\tau = \perp$ or $\tau \vDash \neg q$.

   e. If $u \Leftrightarrow w$, then u is also the weakest precondition of S and q.
      True. For example, if $wp(S, q)$ is $x \geq 1$, then $x > 0$ or $1 \leq x$ can also be used as the weakest precond

- The **weakest liberal precondition for** S **and** q, written $wlp\ (S, q)$, is like $w(S, q)$ but for partial correctness. In other words, $wlp\ (S, q)$ is a valid precondition for q under partial correctness where no weaker valid precondition exists.
  - In symbols: $\big(w = wlp\ (S, q)\big) \Leftrightarrow \vDash \{w\}\ S\ \{q\} \wedge \forall u. (\vDash \{u\}\ S\ \{q\}) \leftrightarrow (u \to w)$.
  - In terms of collection of states: $wlp\ (S, q) = \{\sigma \in \Sigma \mid M(S, \sigma) - \perp \vDash q\}$.

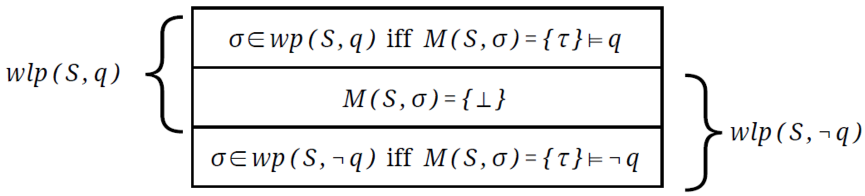
- We care about wp and wlp since they are the most general conditions a program requires to run "successfully" in when we want to get a certain postcondition.
  - From one of the above examples, we learned that if a state σ does not satisfy wp, then it is guaranteed that $M(S, \sigma) \nvDash q$. Similarly, for wlp, if $\sigma \nvDash wlp$, then $M(S, \sigma) - \perp \nvDash q$.

- We talked about two ways to understand $wp(S, q)$: as a predicate and as a collection of states. We keep the second understanding because sometimes we want to say "$wp(S, q) \vDash \cdots$" or "$wp(S, q) \vDash_{tot} \ldots$"; it is also quite convenient to have a set that represents the collection of all possible states that works for S and q.

- Here is one way to connect these two understandings. When we want to express that "σ satisfies $wp(S, q)$":
  - If $wp(S, q)$ is considered as a predicate, then we can say $\sigma \vDash wp(S, q)$.
  - If $wp(S, q)$ is considered as a collection of states, then we can say $\sigma \in wp(S, q)$.

(wp and wlp for deterministic program)

- The following figure illustrates the relationships between wp and wlp for *deterministic* programs. Here it uses the definitions of wp and wlp as they are set of states.



  - For a state σ and a deterministic program S, we can have three possible outcomes for $M(S,\sigma)$:

  1) $M(S,\sigma) = \{\tau\}$ and $\{\tau\} \models q$.
  2) $M(S,\sigma) = \{\perp\}$
  3) $M(S,\sigma) = \{\tau\}$ and $\{\tau\} \models \neg q$.

  - $wp\ (S,q)$ is set of all σ in situation 1).
  - $wp\ (S,\neg q)$ is set of all σ in situation 3).
  - $wlp\ (S,q)$ is set of all σ in situation 1) and 2).
  - $wlp\ (S,\neg q)$ is set of all σ in situation 2) and 3).

3. True or False.
   a. Let S be deterministic. $wlp(S,T) \Leftrightarrow T$.
   True. Because, for any state σ, either $M(S,\sigma) = \perp$ or $M(S,\sigma) \neq \perp$. If $M(S,\sigma) = \perp$, then $\sigma \in wlp(S,T)$; if $M(S,\sigma) \neq \perp$, then $M(S,\sigma) \models T$. Thus, all states are in $wlp(S,T)$; in other words, $wlp(S,T) \Leftrightarrow T$.

   b. Let S be deterministic. $wp(S,F) \Leftrightarrow F$.
   True. For any state σ, either $M(S,\sigma) = \perp$ or $M(S,\sigma) \neq \perp$. If $M(S,\sigma) = \perp$, then $\sigma \notin wp(S,F)$; if $M(S,\sigma) \neq \perp$, then $M(S,\sigma) \not\models F$. Thus, $wp(S,F)$ is an empty set; in other words, $wp(S,F) \Leftrightarrow F$.

   c. $wp(y \coloneqq x * x,\ y \geq 4) \Leftrightarrow wlp(y \coloneqq x * x,\ y \geq 4)$
   True, because the statement $y \coloneqq x * x$ is loop-free and cannot create a runtime error. As an aside, using backward assignment rule, we can get $wp(y \coloneqq x * x,\ y \geq 4) \Leftrightarrow x * x \geq 4$.

   d. $wp(S,q) \Leftrightarrow wlp(S,q)$, where $S \equiv \textbf{while}\ x \neq 0\ \textbf{do}\ x \coloneqq x - 1\ \textbf{od}$, $q \equiv x = 0$.
   False. It is easy to see that $wp(S,q)$ is $x \geq 0$,because $x < 0$ will make S diverges. $wlp(S,q)$ is T, since we can accept S diverges.

(wp and wlp in general programs)

- We need to be careful when nondeterminism is considered, $M(S,\sigma)$ might contain more than one states.
  - $\sigma \in wp(S,q) \Leftrightarrow M(S,\sigma) \models q$.
  - $\sigma \in wlp(S,q) \Leftrightarrow M(S,\sigma) - \perp \models q$.
  - $\sigma \notin wp(S,q) \Leftrightarrow \exists \tau \in M(S,\sigma).\ \tau = \perp \vee \tau \not\models q$.
  - $\sigma \notin wlp(S,q) \Leftrightarrow \exists \tau \in M(S,\sigma).\ \tau \neq \perp \wedge \tau \not\models q$.

4. Show the following property: $wp(S, q_1) \wedge wp\ (S, q_2) \Leftrightarrow wp(S, q_1 \wedge q_2)$.
   - If a state $\sigma \in wp(S, q_1) \wedge wp\ (S, q_2)$, then $\sigma \in wp(S, q_1)$ and $\sigma \in wp(S, q_2)$; then $M(S, \sigma) \vDash q_1$ and $M(S, \sigma) \vDash q_2$; thus $M(S, \sigma) \vDash q_1 \wedge q_2$, which implies $\sigma \in wp(S, q_1 \wedge q_2)$.

   - If a state $\sigma \in wp(S, q_1 \wedge q_2)$, then $M(S, \sigma) \vDash q_1 \wedge q_2$; thus $M(S, \sigma) \vDash q_1$ and $M(S, \sigma) \vDash q_2$, which implies $\sigma \in wp(S, q_1) \wedge wp\ (S, q_2)$.

   - Using a similar proof, we can also show the following property: $wlp(S, q_1) \wedge wlp\ (S, q_2) \Leftrightarrow wlp(S, q_1 \wedge q_2)$.

5. Show the following property: $wp(S, q_1) \vee wp\ (S, q_2) \Rightarrow wp(S, q_1 \vee q_2)$.
   - If a state $\sigma \in wp(S, q_1) \vee wp\ (S, q_2)$, then $\sigma \in wp(S, q_1)$ **or** $\sigma \in wp(S, q_2)$; then $M(S, \sigma) \vDash q_1$ **or** $M(S, \sigma) \vDash q_2$; thus $M(S, \sigma) \vDash q_1 \vee q_2$, which implies $\sigma \in wp(S, q_1 \vee q_2)$.

   - How about the inverse of this property? Is it true that "$wp(S, q_1) \vee wp\ (S, q_2) \Leftarrow wp(S, q_1 \vee q_2)$" ?
     - When $M(S, \sigma) = \{\tau\}$ ($M(S, \sigma)$ contains only one state):
       If $\sigma \in wp(S, q_1 \vee q_2)$, then $M(S, \sigma) \vDash q_1 \vee q_2$, and $M(S, \sigma) \vDash q_1$ or $M(S, \sigma) \vDash q_2$; then $\sigma \in wp(S, q_1)$ or $\sigma \in wp(S, q_2)$ which implies $\sigma \in wp(S, q_1) \vee wp\ (S, q_2)$.

     - When $M(S, \sigma)$ contains more than one states:
       Let $M(S, \sigma) \supseteq \{\tau_1, \tau_2\}$. When $M(S, \sigma) \vDash q_1 \vee q_2$, it is possible that $\tau_1 \vDash q_1$ and $\tau_2 \vDash q_2$. So, even if we can have $M(S, \sigma) \vDash q_1 \vee q_2$, but don't necessarily have $M(S, \sigma) \vDash q_1$ or $M(S, \sigma) \vDash q_2$.

     - To sum up, $wp(S, q_1) \vee wp\ (S, q_2) \Leftarrow wp(S, q_1 \vee q_2)$ is only true when S is deterministic (or $M(S, \sigma)$ contains only one state).

   - Using a similar proof, we can also show the following property: $wlp(S, q_1) \vee wlp\ (S, q_2) \Rightarrow wlp(S, q_1 \vee q_2)$. But $wlp(S, q_1) \vee wlp\ (S, q_2) \Leftarrow wlp(S, q_1 \vee q_2)$ only holds when S is deterministic (or $M(S, \sigma)$ contains only one state).