Weakest Preconditions pt. 1

CS 536: Science of Programming, Fall 2019

9/20

A. Why

- Weakest liberal preconditions (*wlp*) and weakest preconditions (*wp*) are the most general requirements that a program must meet to be correct

B. Objectives

At the end of today you should understand

- What *wlp* and *wp* are and how they are related to preconditions in general.

C. Weaker and Weaker Preconditions?

- Say we have a triple $\models \{p_0\}\ S\ \{q\}$. There may or may not be a strictly weaker precondition we can use instead of p_0. I.e., some p_1 where $\models \{p_1\}\ S\ \{q\}$ with p_1 strictly weaker than p_0? (I.e, $p_1 \rightarrow p_0$, but not vice versa). Similarly, there might be an even strictly weaker p_2 that's valid as a precondition and so on.
  - (Note there's always a not-strictly weaker precondition: For an easy example, take $p_1 \equiv p_0 \wedge \mathtt{T}$, $p_2 \equiv p_1 \wedge \mathtt{T}$, etc.)
- So does the sequence $\ldots, p_2, p_1, p_0$ have to have a beginning? (Or reading the sequence backwards, is there a limit?) If if $\models \{\mathtt{T}\}\ S\ \{q\}$, then the sequence stops, since there's no predicate strictly weaker than true.
- In general, it turns out that there's always a limit to the sequence $\ldots, p_2, p_1, p_0$. We call this limit the **weakest liberal precondition** (*wlp*) of *S* and *q*, written *wlp*(*S*, *q*). This limit is useful because it describes the largest set of states that gives us partial correctness.
- The key here is "largest set". If *w* is the weakest liberal precondition for *S* and *q*, then *no p' strictly weaker than w is a valid precondition for S and q*.
- *wlp* is for partial correctness; for $\models_{tot}$ the notion is *wp*(*S*, *q*), the **weakest precondition** of *S* and *q*.
- **Example**: If $\mathtt{x} \in \{y \in \mathbb{Z} \mid \mathtt{y} \geq 0\}$ then $\{2 \leq \mathtt{x} \leq 6\}\ \mathtt{x} := \mathtt{x*x}\ \{\mathtt{x} \geq 4\}$ is valid, and we can form the sequence $2 \leq \mathtt{x}, \ldots, 2 \leq \mathtt{x} \leq 8, 2 \leq \mathtt{x} \leq 7, 2 \leq \mathtt{x} \leq 6$. Nothing weaker than $2 \leq \mathtt{x}$ is a precondition, so it's the $wp(\mathtt{x} := \mathtt{x*x}, \mathtt{x} \geq 4)$.

D. Notation

- **Notation**: *Sat*(*p*) is the set of states that satisfy *p*: $Sat(p) = \{\sigma \in \Sigma \mid \sigma \models p\}$.
  - (Note some people write $[\![p]\!]$ for *Sat*(*p*).)
- Using this notation, we can say
  - $\sigma \models_{tot} \{p\}\ S\ \{q\}$ iff $M(S, \sigma) \subseteq Sat(q)$.
    - Since $\bot \not\models q$, we can't have $M(S, \sigma) \subseteq Sat(q)$ if $\bot \in M(S, \sigma)$, so this guarantees termination of *S*.
  - $\sigma \models \{p\}\ S\ \{q\}$ iff $M(S, \sigma) - \{\bot\} \subseteq Sat(q)$.
    - The original phrasing was $\sigma \models \{p\}\ S\ \{q\}$ iff $\sigma \models p$ implies $M(S, \sigma) - \{\bot\}$ is $\varnothing$ or $\models q$.

- Using $\subseteq$ covers the case where $M(S, \sigma) = \{\perp\}$ without having to name it explicitly

E. The Weakest Liberal Precondition (wlp) and Weakest Precondition (wp)

- Formally, we can define *wlp* and *wp* using states:
- **Definition**: $wlp(S, q)$, the **weakest liberal precondition** of statement S with respect to a postcondition q, is the set of all states that satisfy $\{p\}\ S\ \{q\}$ for partial correctness. $wlp(S, q)) = \{\sigma \in \Sigma \mid \sigma \vDash \{p\}\ S\ \{q\}\}$
- **Definition**: $wp(S, q)$, the **weakest precondition** of statement S with respect to a postcondition q, is the set of all states that satisfy $\{p\}\ S\ \{q\}$ for total correctness: $wp(S, q) = \{\sigma \in \Sigma \mid \sigma \vDash_{tot} \{p\}\ S\ \{q\}\}$
  - Note $wp(S, q) \rightarrow wlp(S, q)$. If $wlp(S, q) \wedge \neg wp(S, q)$ is satisfiable iff running S under σ might not terminate.
- The important property of *wlp* and *wp* is that any start state outside of them does not satisfy $\{p\}\ S\ \{q\}$ (under partial correctness for *wlp* and total correctness for *wp*).
- We can treat *wlp* and *wp* as yielding a predicate
  - w is a $wlp(S, q)$ iff $Sat(w) = wlp(S, q)$
  - w is a $wp(S, q)$ iff $Sat(w) = wp(S, q)$
- **Note**: w is “a” *wlp/wp* because as any predicate $\Leftrightarrow w$ is also a *wlp/wp*. (Trivial examples are $w \wedge T$, $w \wedge T \wedge T$, etc.) We say that w is determined “up to ” logical equivalence, so “Let w be **the** *wlp*/*wp* of S and q” really means “Let w be any predicate $\Leftrightarrow$ *wlp/wp* of S and q.”
- Now we can rephrase the definitions of *wlp/wp* using predicates:
  - $\vDash \{p\}\ S\ \{q\}$ iff $\vDash p \rightarrow wlp(S, q))$
  - $\vDash_{tot} \{p\}\ S\ \{q\}$ iff $\vDash p \rightarrow wp(S, q))$.
- Equivalent phrasings
  - $\vDash \{wlp(S, q)\}\ S\ \{q\}$ and $\vDash \{p\}\ S\ \{q\}$ iff $\vDash p \rightarrow wlp(S, q))$.
  - If $\vDash p \rightarrow wlp(S, q))$ then $\vDash \{p\}\ S\ \{q\}$, but if $\nvDash p \rightarrow wlp(S, q))$ then $\nvDash \{p\}\ S\ \{q\}$.
- For total correctness,
  - If $\vDash p \rightarrow wp(S, q))$ then $\vDash_{tot} \{p\}\ S\ \{q\}$, but if $\nvDash p \rightarrow wlp(S, q))$ then $\nvDash \{p\}\ S\ \{q\}$.
  - $\vDash_{tot} \{wp(S, q)\}\ S\ \{q\}$ and $\vDash \{p\}\ S\ \{q\} \vDash_{tot} p \rightarrow wp(S, q))$.

F. wp and wlp for Deterministic Programs

- If S is deterministic, then S leads to a unique result: $M(S, \sigma) = \{\tau\}$ for some $\tau \in \Sigma_\perp$.
- If S terminates normally ($\tau \in \Sigma$), then the start state σ is part of either $wlp/wp(S, q)$ or $wlp/wp(S, \neg q)$, depending on whether τ satisfies q or $\neg q$.
- Since $wp(S, q)$ is the set of states that lead to satisfaction of q, $\neg wp(S, q)$ is the set of states that lead to an error or to satisfaction of $\neg q$. Similarly, $\neg wp(S, \neg q)$ is the set of states that lead to an error or to satisfaction of q. The intersection of these two sets, $\neg wp(S, q) \wedge \neg wp(S, \neg q)$, is the set of states that lead to an error.
- Since σ must lead S either to termination satisfying q, termination satisfying $\neg q$, or nontermination, every state satisfies exactly one of $wp(S, q)$, $wp(S, \neg q)$, and $\neg wp(S, q) \wedge \neg wp(S, \neg q)$.
- Let $E \equiv \neg wp(S, q) \wedge \neg wp(S, \neg q)$, then we get the identities

- $\neg\, wp(S, q) \Leftrightarrow E \vee wp(S, \neg\, q)$. The negation of "*S* terminates with *q* true" is "*S* doesn't terminate or it terminates with *q* false".
- $\neg\, wp(S, \neg\, q) \Leftrightarrow E \vee wp(S, q)$ is symmetric: The negation of "*S* terminates with *q* false" is "*S* doesn't terminate or it terminates with *q* true".

- If *S* contains a loop and $M(S, \sigma)$ diverges (and $\sigma \in \Sigma$), then $\sigma \vDash \neg\, wp(S, q) \wedge \neg\, wp(S, \neg\, q)$. (See Figure 3.)
- On the left of Figure 3 is the set of all states Σ broken up into three partitions
  - The states that establish *q* form $wp(S, q) = \{\sigma \in \Sigma \mid M(S, \sigma) = \{\tau\} \text{ and } \tau \vDash q\}$
  - The states that establish $\neg\, q$ form $wp(S, \neg\, q) = \{\sigma \in \Sigma \mid M(S, \sigma) = \{\tau\} \text{ and } \tau \vDash \neg\, q\}$
  - The states that lead to $\perp$ form $\neg\, wp(S, q) \wedge \neg\, wp(S, \neg\, q) = \{\sigma \in \Sigma \mid M(S, \sigma) = \{\perp\}\}$
- The arrows indicate that starting from $wp(S, q)$ or $wp(S, \neg\, q)$ yields a state that satisfies *q* or $\neg\, q$ respectively. Starting from a state outside both weakest preconditions leads to an error.

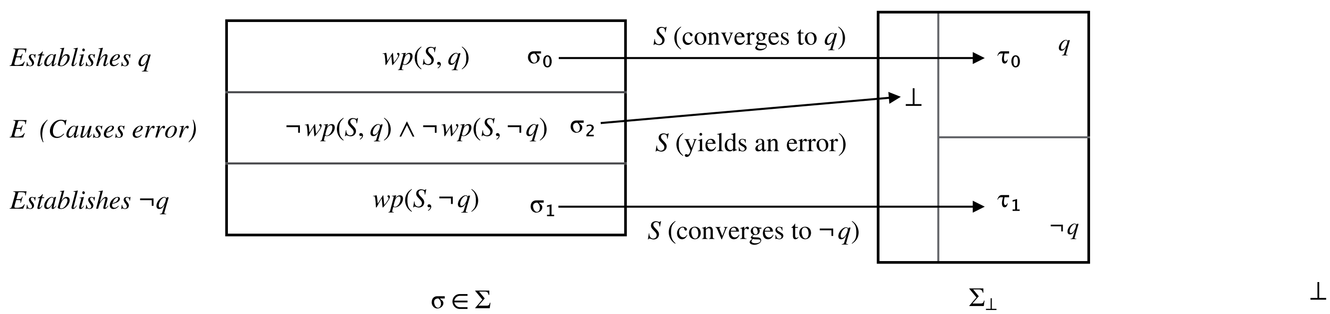


Figure 3: The Weakest Precondition for Deterministic S

$M(S, \sigma_0) = \{\tau_0\} \vDash q$

$M(S, \sigma_1) = \{\tau_1\} \vDash \neg\, q$

$M(S, \sigma_2) = \{\perp\} \nvDash q$ and $\nvDash \neg\, q$

- Figure 4 shows how that with deterministic programs, the $wlp(S, q)$ combines $wp(S, q)$ with the states that cause errors; similarly, the $wlp(S, \neg\, q)$ combines $wp(S, \neg\, q)$ with the states that cause errors.
- I.e., $\sigma \vDash wlp(S, q)$ when $M(S, \sigma) - \{\perp\} \subseteq Sat(q)$, $\sigma \vDash wlp(S, \neg\, q)$ when $M(S, \sigma) - \{\perp\} \subseteq Sat(\neg q)$. (Note this allows for $M(S, \sigma) = \{\perp\}$ without naming it as a special case.)

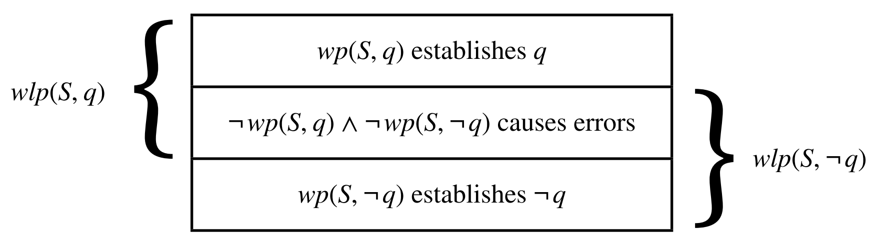
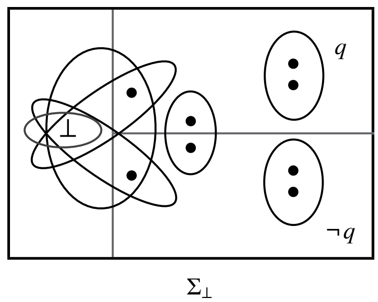


Figure 4: The Weakest Liberal Precondition for Deterministic S

G. *wp and wlp for Nondeterministic Programs*

- If S is nondeterministic, then $M(S, \sigma)$ is a nonempty subset of $\Sigma_\perp$ that can contain more than one member. To satisfy q or $\neg q$, all the states in then $M(S, \sigma)$ must satisfy q or $\neg q$ respectively.
- Figure 5 shows the possible situations:
  - $\sigma \vDash wp(S, q)$ when everything in $M(S, \sigma)$ satisfies q.
  - $\sigma \vDash wp(S, \neg q)$ if everything in $M(S, \sigma)$ satisfies $\neg q$,
  - $\sigma \vDash \neg wp(S, q)$ when $\perp \in M(S, \sigma)$ and/or $\tau \vDash \neg q$ for some $\tau \in M(S, \sigma)$.
  - $\sigma \vDash \neg wp(S, \neg q)$ when $\perp \in M(S, \sigma)$ and/or $\tau \vDash q$ for some $\tau \in M(S, \sigma)$.
  - $\sigma \vDash wp(S, q) \wedge \neg wp(S, \neg q)$ when $\perp \in M(S, \sigma)$ and/or $\tau_1 \vDash q$ and $\tau_2 \vDash \neg q$ for some $\{\tau_1, \tau_2\} \subseteq M(S, \sigma)$.

On the left: $M(S, \sigma)$ when $\sigma \vDash \neg wp(S, q) \wedge \neg wp(S, \neg q)$



On the right: $M(S, \sigma)$ when $\sigma \vDash wp(S, q)$ or $\sigma \vDash wp(S, \neg q)$

Figure 5: Weakest Precondition $M(S, \sigma)$ for Non-Deterministic S

- For non-deterministic programs, the situation for $wlp(S, q)$ is similar to the situation for deterministic programs in that $\sigma \vDash wlp(S, q)$ when $M(S, \sigma) - \{\perp\} \subseteq Sat(q)$. In Figure 6, the $wlp(S, q)$ is satisfied by σ that lead to the top or middle sets, and the the $wlp(S, \neg q)$ is satisfied by σ that lead to the middle or bottom sets.

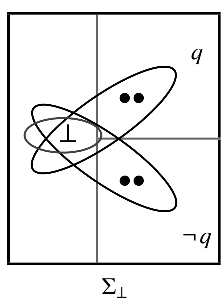


Figure 6: Weakest Liberal Preconditions $M(S, \sigma)$ for Non-Deterministic S

- Finally, Figure 7 shows how with nondeterministic programs, starting S outside the weakest precondition for q can still terminate in a state satisfying q: Even for $\sigma \nvDash wp(S, q)$ where $\perp \notin M(S, \sigma)$, it's possible for $M(S, \sigma) \cap Sat(q) \neq \varnothing$ because $M(S, \sigma) \cap Sat(\neg q)$ also $\neq \varnothing$.
- **Example 9**: Let $S \equiv$ **if** $x \geq 0 \rightarrow x := 10 \;\square\; x \leq 0 \rightarrow x := 20$ **fi**, and let $\Sigma_0 = M(S, \{x = 0\})$ be the set with two states $\{\{x = 10\}, \{x = 20\}\}$. Then $\Sigma_0 \nvDash x = 10$, $x \neq 10$, $x = 20$, and $x \neq 20$. (We do have $\Sigma_0 \vDash x = 10 \vee x = 20$.)

- --- ended here 2019-09-18

H. Disjunctive Postconditions

- There are some relationships that hold between the *wp* of a predicate and the *wp*'s of its subpredicates.
- E.g., if you start in a state that is guaranteed to lead to a result that satisfies q_1 and q_2 separately, then the result will also satisfy $q_1 \wedge q_2$, and vice versa. In symbols, $wp(S, q_1) \wedge wp(S, q_2) \Leftrightarrow wp(S, q_1 \wedge q_2)$.
  - This relationship holds for both deterministic and nondeterministic *S*.
  - The relationship between $wp(q_1 \vee q_2)$ and $wp(q_1)$ and $wp(q_2)$ differs for deterministic and nondeterministic *S*.
- Deterministic *S*: For all *S*, $wp(S, q_1) \vee wp(S, q_2) \Leftrightarrow wp(S, q_1 \vee q_2)$
  - Nondeterministic *S*: For all *S*, $wp(S, q_1) \vee wp(S, q_2) \Rightarrow wp(S, q_1 \vee q_2)$, but $\Leftarrow$ doesn't hold for some *S*.
  - For deterministic *S*, $M(S, \sigma) = \{\tau\}$ for some $\tau \in \Sigma_\perp$. If $\tau \vDash q_1 \vee q_2$ then either $\tau \vDash q_1$ or $\tau \vDash q_2$ (or both).
- So if $M(S, \sigma) \neq \{\perp\}$, then $M(S, \sigma) \vDash q_1 \vee q_2$ iff $M(S, \sigma) \vDash q_1$ or $M(S, \sigma) \vDash q_2$.
  - Because of this, $wp(S, q_1) \vee wp(S, q_2) \Leftrightarrow wp(S, q_1 \vee q_2)$.
  - For nondeterministic *S*, we still have $wp(S, q_1) \vee wp(S, q_2) \Rightarrow wp(S, q_1 \vee q_2)$. I.e., if you start in a state that's guaranteed to terminate satisfying q_1, or guaranteed to terminate satisfying q_2, then that state is guaranteed to terminate satisfying $q_1 \vee q_2$.
- For nondeterministic *S*, the other direction, $wp(S, q_1) \vee wp(S, q_2) \Leftarrow wp(S, q_1 \vee q_2)$, doesn't always hold: *S* can guarantee establishing $q_1 \vee q_2$ without leaving any way to guarantee satisfaction of just q_1 or just q_2.
- **Example 10**: Let $CoinFlip \equiv$ **`if`** `T` → `x := 0` □ `T` → `x := 1` **`fi`**.
  - For all σ, $M(CoinFlip, \sigma) = \{\{x = 0, x = 1\}\}$, which $\vDash x = 0 \vee x = 1$ but $\nvDash x = 0$ and $\nvDash x = 1$.
  - Let $Heads \Leftrightarrow wp(CoinFlip, x = 0)$, $Tails = wp(CoinFlip, x = 1)$, and $Heads\_or\_Tails = wp(CoinFlip, x = 0 \vee x = 1)$. We find $Heads \Leftrightarrow Tails \Leftrightarrow$ F but $Heads\_or\_Tails \Leftrightarrow$ T.
  - Altogether, $(Heads \vee Tails) \Rightarrow$ (but not $\Leftarrow$) $Heads\_or\_Tails$.
- So for nondeterministic *S*, even though $\vDash_{tot} \{wp(S, q)\}\ S\ \{q\}$, if *q* is disjunctive, it's possible for you to run *S* in a state $\sigma \vDash \neg wp(S, q)$ but still terminate without error in a state satisfying *q*. (For deterministic *S*, this won't happen.) E.g., if $S \equiv$ **`if`** *B* **`then`** `x := 0` **`else`** `x := 1` **`fi`**, then $M(S, \sigma) \vDash x = 0$ or $M(S, \sigma) \vDash x = 1$ (tails), $wp(S, x = 0) \Leftrightarrow B$ and $wp(S, x = 1) \Leftrightarrow \neg B$.

I. The Weakest Liberal Precondition (wlp)

- The relationship between the **weakest precondition** (*wp*) and the **weakest liberal precondition** (*wlp*) is the same as total vs partial correctness.
  - $wp(S, q)$ is the set of start states that guarantee termination establishing *q*.
  - $wlp(S, q)$ is the set of start states that guarantee (causing an error or termination establishing *q*).
- **Definition**: The **weakest liberal precondition** of *S* and *q*, written $wlp(S, q)$, is the predicate *w* such that $\vDash \{w\}\ S\ \{q\}$ and for every $\sigma \vDash \neg w$, $\perp \notin M(S, \sigma)$ and $M(S, \sigma) \nvDash q$.

- If we start in a state σ satisfying *wlp*(S, q) then either some execution path for S in σ causes an error or else all execution paths for S in σ lead to final states that ⊨ q. If we start in a σ satisfying ¬*wlp*(S, q), then every execution path for S in σ leads to a final state and at least one of the final states ⊨ ¬q.
- We always have *wp*(S, q) ⇒ *wlp*(S, q); the other direction, *wp*(S, q) ⇐ *wlp*(S, q), only holds if S never causes an error.
- **Example 11**: Let $W \equiv$ **while** $x \neq 0$ **do** $x := x-1$**;** $y := 0$ **od**, then for $M(W, \sigma)$,
  - If $\sigma \vDash x = 0$ then $M(W, \sigma) = \{\sigma\}$. Note if $\sigma \vDash x = 0 \wedge y = 0$ then $M(W,\sigma) = \{\sigma\}$
  - If $\sigma \vDash x > 0$ then $M(W, \sigma) = \{\sigma[x \mapsto 0][y \mapsto 0]\}$
    - Note the only way W terminates with $y \neq 0$ is if we run it in $x = 0 \wedge y \neq 0$.
  - If $\sigma \vDash x < 0$ then $M(W, \sigma) = \{\perp\}$ so for any postcondition q, $x < 0 \rightarrow wlp(W, r)$ and $x < 0 \rightarrow \neg wp(W, q)$.
  - If we look at a particular postcondition, say $q \equiv x = 0 \wedge y = 0$, we find $wlp(W, q) \Leftrightarrow x > 0 \vee x = y = 0 \vee x < 0$ and $wp(W, q) \Leftrightarrow x > 0 \vee x = y = 0$. For $\neg q \Leftrightarrow x \neq 0 \vee y \neq 0$, since W can never terminate with $x \neq 0$, we find $wlp(W, \neg q) \Leftrightarrow wp(W, y \neq 0) \Leftrightarrow x = 0 \wedge y \neq 0 \vee x < 0$ and $wp(W, \neg q) \Leftrightarrow wp(W, y \neq 0) \Leftrightarrow x = 0 \wedge y \neq 0$.
- The "being weakest" property of *wlp* is similar to that for *wp*, but for partial correctness: ⊨ $\{wlp(S, q)\}\ S\ \{q\}$ and for all p, ⊨ $\{p\}\ S\ \{q\}$ iff ⊨ $p \rightarrow wlp(S, q)$.

J. *Calculating wlp for Loop-Free Programs*

- It's easy to calculate the *wlp* of a loop-free program.
  - If a loop-free program cannot cause a runtime error then its *wp* and *wlp* are the same, which is also nice.
- The following algorithm takes S and q where S has no loops and syntactically calculates a particular predicate for $wlp(S, q)$, which is why it's described using $wlp(S, q) \equiv \ldots$ instead of $wp(S, q) \Leftrightarrow \ldots$.
  - $wlp(\textbf{skip}, q) \equiv q$
  - $wlp(v := e,\ Q(v)) \equiv Q(e)$ where Q is a predicate function over one variable
    - The operation that takes us from $Q(v)$ to $Q(e)$ is called **syntactic substitution**; we'll look at it in more detail soon, but in the simple case, we simply inspect the definition of Q, searching its text for occurrences of the variable v and replacing them with copies of e.
  - $wlp(S_1\ \textbf{;}\ S_2, q) \equiv wlp(S_1, wlp(S_2, q))$
  - $wlp(\textbf{if}\ B\ \textbf{then}\ S_1\ \textbf{else}\ S_2\ \textbf{fi}, q) \equiv (B \rightarrow w_1) \wedge (\neg B \rightarrow w_2)$ where $w_1 \equiv wlp(S_1, q)$ and $w_2 \equiv wlp(S_2, q)$. If you want, you can write $(B \wedge w_1) \vee (\neg B \wedge w_2)$, which is equivalent.
  - $wlp(\textbf{if}\ B_1 \rightarrow S_1 \ \square\ B_2 \rightarrow S_2\ \textbf{fi} \equiv (B_1 \rightarrow w_1) \wedge (B_2 \rightarrow w_2)$ where $w_1 \equiv wlp(S_1, q)$ and $w_2 \equiv wlp(S_2, q)$.
    - For the nondeterministic **if**, don't write $(B_1 \wedge w_1) \vee (B_2 \wedge w_2)$ instead of $(B_1 \rightarrow w_1) \wedge (B_2 \rightarrow w_2)$; they aren't logically equivalent. When B_1 and B_2 are both true, either S_1 or S_2 can run, so we need $B_1 \wedge B_2 \rightarrow w_1 \wedge w_2$.
    - Using $(B_1 \wedge w_1) \vee (B_2 \wedge w_2)$ fails because it allows for the possibility that B_1 and B_2 are both true but one of w_1 and w_2 is not true. This isn't a problem when $B_2 \Leftrightarrow \neg B_1$, which is why we can use $(B \wedge w_1) \vee (\neg B \wedge w_2)$ with deterministic **if** statements.

Strength; Weakest Preconditions, pt. 1

CS 536: Science of Programming

A. Why

- The weakest precondition and weakest liberal preconditions are the most general preconditions that a program needs in order to run correctly.

B. Objectives

At the end of this activity you should be able to

- Define what a weakest liberal precondition (*wlp*) and weakest precondition (*wp*) is and how it's related to (and different from) preconditions in general
- Be able to calculate the *wlp* of a simple loop-free program.

C. Problems

1. Let $w \Leftrightarrow wp(S, q)$ and let S be deterministic.
   a. For which $\sigma \vDash w$ do we have $\sigma \vDash_{tot} \{w\}\ S\ \{q\}$?
   b. For which $\sigma \vDash \neg w$ do we have $\sigma \vDash \{\neg w\}\ S\ \{q\}$?
   c. For which $\sigma \vDash w$ do we have $\sigma \vDash_{tot} \{w\}\ S\ \{\neg q\}$?
   d. For which $\sigma \vDash \neg w$ do we have $\sigma \vDash \{\neg w\}\ S\ \{\neg q\}$?
   e. If S is nondeterministic, how do we have to modify the statement in part (d)?

2. If $\sigma \vDash w$ and $\sigma \vDash \{w\}\ S\ \{q\}$ and $\sigma \nvDash_{tot} \{w\}\ S\ \{q\}$,
   a. What can we conclude about $M(S, \sigma)$?
   b. If in addition, S is deterministic, what more can we conclude about $M(S, \sigma)$?

3. For an arbitrary p (not necessarily one that implies w), what $\vDash$ and $\vDash_{tot}$ properties relationships do the triples
   a. $\{p \wedge w\}\ S\ \{q\}$ and $\{\neg p \wedge w\}\ S\ \{q\}$ have?
   b. $\{p \wedge \neg w\}\ S\ \{\neg q\}$ and $\{\neg p \wedge \neg w\}\ S\ \{\neg q\}$ have, if S is deterministic?
   c. $\{p \wedge \neg w\}\ S\ \{q\}$ and $\{\neg p \wedge \neg w\}\ S\ \{q\}$ have, if S is nondeterministic?

4. How are $wp(S, q_1 \vee q_2)$ and $wp(S, q_1) \cup wp(S, q_2)$, related if S is deterministic? If S is nondeterministic?

5. Which of the following statements are correct ?

   a. For all $\sigma \in \Sigma$, $\sigma \vDash wp(S, q)$ iff $M(S, \sigma) \vDash q$

   b. For all $\sigma \in \Sigma$, $\sigma \vDash wlp(S, q)$ iff $M(S, \sigma) \cup \Sigma \vDash q$

   c. $\vDash_{tot} \{wp(S, q)\}\ S\ \{q\}$

   d. $\vDash \{wlp(S, q)\}\ S\ \{q\}$

   e. $\vDash_{tot} \{p\}\ S\ \{q\}$ iff $\vDash p \rightarrow wp(S, q)$

   f. $\vDash \{p\}\ S\ \{q\}$ iff $\vDash p \rightarrow wlp(S, q)$

   g. $\vDash \{\neg wp(S, q)\}\ S\ \{\neg q\}$

   h. $\vDash_{tot} \{\neg wlp(S, q)\}\ S\ \{\neg q\}$

   i. $wlp(S, q) \wedge wlp(S, \neg q)$ is not satisfiable

   j. $\nvDash p \rightarrow wp(S, q)$ iff $\nvDash_{tot} \{p\}\ S\ \{q\}$

   k. $\nvDash p \rightarrow wlp(S, q)$ iff $\nvDash \{p\}\ S\ \{q\}$

Solution to Activity 10 (Weakest Preconditions, pt. 1)

1. (Properties of weakest preconditions)

   a. For all $\sigma \vDash w$, we have $\sigma \vDash_{tot} \{w\}\ S\ \{q\}$, since w is a precondition for $\vDash_{tot} \{\ldots\}\ S\ \{q\}$.

   b. For no $\sigma \vDash \neg w$ do we have $\sigma \vDash \{\neg w\}\ S\ \{q\}$ because for w to be the weakest precondition for S and q, it cannot be that $M(S, \sigma) \vDash q$.

   c. For no $\sigma \vDash w$ do we have $\sigma \vDash_{tot} \{w\}\ S\ \{\neg q\}$ because w is a precondition for $\vDash_{tot} \{\ldots\}\ S\ \{q\}$.

   d. For all $\sigma \vDash \neg w$, we have $\sigma \vDash \{\neg w\}\ S\ \{\neg q\}$ because for w to be the weakest precondition for S and q, $\sigma \vDash \neg w$ implies $M(S, \sigma) \nvDash q$. Since S is deterministic, either $M(S, \sigma) = \{\perp\}$ or $M(S, \sigma) \vDash \neg q$. Either way, $\sigma \vDash \{\neg w\}\ S\ \{\neg q\}$.

   e. If S is nondeterministic and $M(S, \sigma) \nvDash q$, then as in the deterministic case, nontermination is a possibility ($\perp \in M(S, \sigma)$ can happen). Regardless, we no longer know $M(S, \sigma) \vDash \neg q$ because we can have $M(S, \sigma) \nvDash q$ and $M(S, \sigma) \nvDash \neg q$ simultaneously.

2. (Partial but not total correctness when the *wp* is satisfied)

   a. If $\sigma \vDash w$ and $\sigma \vDash \{w\}\ S\ \{q\}$ then $M(S, \sigma) - \{\perp\} \vDash q$. If $\sigma \nvDash_{tot} \{w\}\ S\ \{q\}$ then $M(S, \sigma) \nvDash q$. This can only happen if $\perp \in M(S, \sigma)$. (I.e., S can diverge under σ.)

   b. If in addition S is deterministic, then we don't just have $\perp \in M(S, \sigma)$, we have $\{\perp\} = M(S, \sigma)$. (I.e., S diverges under σ.)

3. (Intersection with *wp*)

   a. $\vDash_{tot} \{p \wedge w\}\ S\ \{q\}$ and $\vDash_{tot} \{\neg p \wedge w\}\ S\ \{q\}$ follow from w being a precondition under $\vDash_{tot}$.

   b. Because w is weakest, we have for all $\sigma \vDash p \wedge \neg w$, that $\sigma \nvDash_{tot}$. $\{p \wedge \neg w\}\ S\ \{q\}$. If S is deterministic, this implies $\sigma \vDash \{p \wedge \neg w\}\ S\ \{\neg q\}$. Similarly, for all $\sigma \vDash \neg p \wedge \neg w$, we have $\sigma \vDash \{p \wedge \neg w\}\ S\ \{\neg q\}$.

   c. If S is nondeterministic then if $\sigma \vDash p \wedge \neg w$, we still know $\sigma \nvDash_{tot}$. $\{p \wedge \neg w\}\ S\ \{q\}$ but both $\sigma \vDash$ and $\sigma \nvDash$ $\{p \wedge \neg w\}\ S\ \{\neg q\}$ are possible. Similarly, if $\sigma \vDash \neg p \wedge \neg w$, we know $\sigma \nvDash_{tot}$. $\{\neg p \wedge \neg w\}\ S\ \{q\}$, but both $\sigma \vDash$ and $\sigma \nvDash$ $\{p \wedge \neg w\}\ S\ \{\neg q\}$ are possible.

4. For deterministic S, $wp(S, q_1 \vee q_2) = wp(S, q_1) \cup wp(S, q_2)$. For nondeterministic S, we have $\supseteq$ instead of $=$.

5. (Properties of *wp* and *wlp*) The following properties are correct:

   (a) and (b) are the basic definitions of *wp* and *wlp*

   (c) and (d) say that *wp* and *wlp* are preconditions

   (e) and (f) say that *wp* and *wlp* are weakest preconditions

   (g) and (h) also say that *wp* and *wlp* are weakest

   (j) and (k) are the contrapositives of (e) and (f).

   However, (i) is incorrect: It claims that $wlp(S, q) \wedge wlp(S, \neg q)$ is never satisfiable, but if $M(S, \sigma) \subseteq \{\perp\}$, then σ satisfies both $wlp(S, q)$ and $wlp(S, \neg q)$.