HOARE LOGIC

VERIFICATION CONDITION GENERATION

- We have already seen that the weakest pre-condition operator can be used to prove Hoare Triples:
  - $\{P\}c\{Q\}$ iff $P \Rightarrow wp(Q, c)$
- Finding exact *wp* for loops is hard. We will instead use the loop invariant as an approximate *wp*.
  - $awp(Q, \text{while(F)@I do c}) = \text{I}$
  - Does this always hold?
- Also need to show that following side-conditions hold:
  - $\{\text{I} \wedge \text{F}\}\text{c}\{\text{I}\}$
  - $\text{I} \wedge \neg\text{F} \Rightarrow Q$

RELATION BETWEEN AWP AND WP

- Let us formally define awp:
  - $\forall \sigma \in awp(Q, c) \,.\, \forall \sigma' \,.\, (\sigma, c) \hookrightarrow^* (\sigma', skip) \rightarrow \sigma' \in Q$
  - Homework: Prove that this holds for $awp(Q,$ while(F)@I do c) = I, when the side-conditions hold.
- We defined $wp(Q, c) \triangleq \{\sigma \mid \forall \sigma' \,.\, (\sigma, c) \hookrightarrow^* (\sigma', \text{skip}) \rightarrow \sigma' \in Q\}$
  - $awp(Q, c) \subseteq wp(Q, c)$

RELATION BETWEEN AWP AND WP

- Let us formally define *awp*:
  - $\forall \sigma \in awp(Q, c)\,.\, \forall \sigma'\,.\, (\sigma, c) \hookrightarrow^* (\sigma', skip) \rightarrow \sigma' \in Q$
  - Homework: Prove that this holds for $awp(Q,$ while(F)@I do c) = I, when the side-conditions hold.
- We defined $wp(Q, c) \triangleq \{\sigma \mid \forall \sigma'\,.\, (\sigma, c) \hookrightarrow^* (\sigma', \text{skip}) \rightarrow \sigma' \in Q\}$
  - $awp(Q, c) \subseteq wp(Q, c)$
- $awp(i \geq 0,$ while(i < n)@(i >= 0) do i := i+1;) = ???

RELATION BETWEEN AWP AND WP

- Let us formally define *awp*:
  - $\forall \sigma \in awp(Q, c) \,.\, \forall \sigma' \,.\, (\sigma, c) \hookrightarrow^* (\sigma', skip) \rightarrow \sigma' \in Q$
  - Homework: Prove that this holds for $awp(Q,$ while(F)@I do c) = I, when the side-conditions hold.
- We defined $wp(Q, c) \triangleq \{\sigma \mid \forall \sigma' \,.\, (\sigma, c) \hookrightarrow^* (\sigma', \text{skip}) \rightarrow \sigma' \in Q\}$
  - $awp(Q, c) \subseteq wp(Q, c)$
- $awp(i \geq 0,$ while(i < n)@(i >= 0) do i := i+1;) = $i \geq 0$

RELATION BETWEEN AWP AND WP

- Let us formally define awp:
  - $\forall \sigma \in awp(Q, c) \,.\, \forall \sigma' \,.\, (\sigma, c) \hookrightarrow^* (\sigma', skip) \rightarrow \sigma' \in Q$
  - Homework: Prove that this holds for $awp(Q,$ while(F)@I do c) = I, when the side-conditions hold.
- We defined $wp(Q, c) \triangleq \{\sigma \mid \forall \sigma' \,.\, (\sigma, c) \hookrightarrow^* (\sigma',$ skip$) \rightarrow \sigma' \in Q\}$
  - $awp(Q, c) \subseteq wp(Q, c)$
- $awp(i \geq 0,$ while(i < n)@(i >= 0) do i := i+1;) = $i \geq 0$
  - $wp(i \geq 0,$ while(i < n)@(i >= 0) do i := i+1;) = ???

RELATION BETWEEN AWP AND WP

- Let us formally define awp:
  - $\forall \sigma \in awp(Q, c) \,.\, \forall \sigma' \,.\, (\sigma, c) \hookrightarrow^* (\sigma', skip) \rightarrow \sigma' \in Q$
  - Homework: Prove that this holds for $awp(Q,$ while(F)@I do c) = I, when the side-conditions hold.
- We defined $wp(Q, c) \triangleq \{\sigma \mid \forall \sigma' \,.\, (\sigma, c) \hookrightarrow^* (\sigma', \text{skip}) \rightarrow \sigma' \in Q\}$
  - $awp(Q, c) \subseteq wp(Q, c)$
- $awp(i \geq 0,$ while(i < n)@(i >= 0) do i := i+1;) = $i \geq 0$
  - $wp(i \geq 0,$ while(i < n)@(i >= 0) do i := i+1;) =$n \geq 0 \vee i \geq 0$

VC GENERATION - I

- We define $VC(Q, c)$ to collect the side-conditions needed for verifying that Q holds after execution of c.
- For while(F)@I do c, there are two side-conditions:
  - $\{\mathsf{I} \wedge \mathsf{F}\}\mathsf{c}\{\mathsf{I}\}$
  - $\mathsf{I} \wedge \neg \mathsf{F} \Rightarrow Q$
- $\{\mathsf{I} \wedge \mathsf{F}\}\mathsf{c}\{\mathsf{I}\}$ is valid if $\mathsf{I} \wedge \mathsf{F} \Rightarrow awp(\mathsf{I}, \mathsf{c})$.
  - c may contain loops, so we also need to consider $VC(\mathsf{I}, \mathsf{c})$.
- Hence,
  $VC(\mathsf{Q}, \text{while(F)@I do c}) \triangleq (\mathsf{I} \wedge \neg \mathsf{F} \Rightarrow \mathsf{Q}) \wedge (\mathsf{I} \wedge \mathsf{F} \Rightarrow awp(\mathsf{I}, \mathsf{c})) \wedge VC(\mathsf{I}, \mathsf{c})$

VC GENERATION - II

- $VC(Q, \text{x:=e}) \triangleq true$
    - Also defined as *true* for all simple program commands (assert, assume, havoc).
- $VC(Q, c_1; c_2) \triangleq ???$

VC GENERATION - II

- $VC(Q, \text{x:=e}) \triangleq true$
    - Also defined as *true* for all simple program commands (assert, assume, havoc).
- $VC(Q, c_1; c_2) \triangleq VC(Q, c_2) \land VC(awp(Q, c_2), c_1)$

VC GENERATION - II

- $VC(Q, \text{x:=e}) \triangleq true$
  - Also defined as *true* for all simple program commands (assert, assume, havoc).
- $VC(Q, c_1; c_2) \triangleq VC(Q, c_2) \wedge VC(awp(Q, c_2), c_1)$
- $VC(Q, \text{if(F) then c}_1 \text{ else c}_2) \triangleq ???$

VC GENERATION - II

- $VC(Q, \text{x:=e}) \triangleq true$
  - Also defined as *true* for all simple program commands (assert, assume, havoc).
- $VC(Q, c_1; c_2) \triangleq VC(Q, c_2) \wedge VC(awp(Q, c_2), c_1)$
- $VC(Q, \text{if(F) then } \mathrm{c}_1 \text{ else } \mathrm{c}_2) \triangleq VC(Q, \mathrm{c}_1) \wedge VC(Q, \mathrm{c}_2)$

VC GENERATION - III

- $awp(Q, c) \triangleq wp(Q, c)$ except for while loops, for which $awp(Q,$ while(F)@I do c) = I.
- Putting it all together, $\{P\}c\{Q\}$ is valid if the following FOL formula is valid:
  - $(P \rightarrow awp(Q, c)) \land VC(Q, c)$

RELATION BETWEEN AWP AND HOARE TRIPLES

- What is the relation between $awp(Q, c)$ and validity of the Hoare Triple $\{P\}c\{Q\}$?
  - Is it possible that $P \to awp(Q, c)$ is valid and $\{P\}c\{Q\}$ is not valid?
  - Is it possible that $\{P\}c\{Q\}$ is valid and $\neg(P \to awp(Q, c))$ is satisfiable?
  - How about $\neg(P \to wp(Q, c))$?

VC GENERATION

SOUNDNESS AND COMPLETENESS

- Is the VC generation procedure sound?
  - Yes. Prove this!
- Is the VC generation procedure complete?
  - No. It is not even relatively complete.
  - The annotated loop invariant may not be strong enough.
- Can the VC generation procedure be fully automated?
  - Yes. Whole point of the exercise!