# CS536, Spring 2022
# Midterm Exam #2
# SOLUTIONS: DO NOT DISTRIBUTE

**Name** ______________________________

**IIT Email** ______________________________

**Important notes:**

- This exam has 15 pages. Make sure you have them all.
- You have 75 minutes to complete the exam. We suggest looking through the questions first to see where to focus your time.
- Use only **blue or black pen** to complete this exam. If you don't have one, ask.
- Write your answers in the space provided. If you need more space for answers or scrap, you can use the back of the page, but clearly mark where your answers are.
- The last 5 pages of this exam are reference material. You may (carefully) tear them off and use them durng the exam. We do not need to collect these pages.
- You are permitted to refer to **two** double-sided 8.5" $\times$ 11" sheets of notes. We will collect your sheets of notes at the end of the exam, so if you want them back, please make sure your full name is clearly written on both pages (or the first one if they're attached). **No other outside aids (including electronics) or notes are permitted.**
- Sign the statement below:
  I have not used any unauthorized resources or received or given help during this exam.

Signed______________________________ Date____________________

| **Question** | 1 | 2 | 3 | 4 | 5 | **Total** |
|---|---|---|---|---|---|---|
| **Points** | 14 | 18 | 30 | 24 | 14 | 100 |
| **Score** | | | | | | |
| **Grader** | | | | | | |

## 1 Expanding Proof Outlines (14 points)

Expand the proof outline below into a full proof outline. **List the resulting proof obligations.** You **do not** need to prove these obligations (but they should be true statements!)

$$\{i \neq j \wedge m \neq i \wedge m \neq j \wedge a[j] \neq a[i]\}$$

if $(a[j] < a[i])$ then $\{$

$a[m] := a[j]$

$\}$ else $\{$

$a[m] := a[i]$

$\}$

$$\{a[m] < a[i] \vee a[m] < a[j]\}$$

| | |
|---|---|
| | $\{i \neq j \wedge m \neq x \wedge a[j] \neq a[i]\}$ |
| if $(a[j] < a[i])$ then $\{$ | $\{i \neq j \wedge m \neq i \wedge m \neq j \wedge a[j] \neq a[i] \wedge a[j] < a[i]\} \Rightarrow \{a[j] < (i = m \, ? \, a[j] : a[i])\}$ |
| $a[m] := a[j]$ | $\{a[m] < a[i]\}$ |
| $\}$ else $\{$ | $\{i \neq j \wedge m \neq i \wedge m \neq j \wedge a[j] \neq a[i] \wedge a[j] \geq a[i]\} \Rightarrow \{a[i] < (j = m \, ? \, a[i] : a[j])\}$ |
| $a[m] := a[i]$ | $\{a[m] < a[j]\}$ |
| $\}$ | $\{a[m] < a[i] \vee a[m] < a[j]\}$ |

Obligations: $\{i \neq j \wedge m \neq i \wedge m \neq j \wedge a[j] \neq a[i] \wedge a[j] < a[i]\} \Rightarrow \{a[j] < (i = m \, ? \, a[j] : a[i])\}$
$\{i \neq j \wedge m \neq i \wedge m \neq j \wedge a[j] \neq a[i] \wedge a[j] \geq a[i]\} \Rightarrow \{a[i] < (j = m \, ? \, a[i] : a[j])\}$

## 2 Concepts (18 points)

Answer each question in 2-5 English sentences. Be as clear and to-the-point as possible.

(a) (6 points) If $s$ has no loops, and $\models \{p\}\ s\ \{q\}$, is it always the case that $\models [\ p \wedge D(s)\ ]\ s\ [\ q\ ]$? Why or why not? If this is only sometimes the case, say when and briefly explain why.

This is the case for loop-free programs; for such programs, if $D(s)$ holds, then there's no error and total correctness holds.

(b) (6 points) Consider the following triple, where we've left the condition and body of the loop unspecified.

$$\{T\}\ \textsf{while}\ e\ \{s\}\ \{x = P(1, n) \wedge a[n] \neq y\}$$

Suggest at least two predicates (write down the predicates) we might want to consider as loop invariants, and explain briefly how you got each one.

$x = P(i, n) \wedge a[n] \neq y$ (replace constant with a variable)
$x = P(1, i) \wedge a[i] \neq y$ (replace constant with a variable)
$x = P(1, n)$ (delete a conjunct)

Consider the following triple, where we've left the condition and body of the loop unspecified.

$$\{T\}\ \mathsf{while}\ e\ \{s\}\ \{x = P(0, n) \wedge a[n] > z\}$$

Suggest at least two predicates (write down the predicates) we might want to consider as loop invariants, and explain briefly how you got each one.

$x = P(i, n) \wedge a[n] > z$ (replace constant with a variable)
$x = P(0, i) \wedge a[i] > z$ (replace constant with a variable)
$x = P(0, n)$ (delete a conjunct)

(c) (6 points) You're trying to find the weakest precondition of the following program, whose postcondition is $x = y$:

$$\mathsf{if}\ x < y\ \mathsf{then}\ \{x := y\}\ \mathsf{else}\ \{\mathsf{skip}\}$$

Chaoqi tells you the weakest precondition is $x \leq y$. Zhenghao tells you it's $(x \geq y \rightarrow x = y)$. Then Stefan says they're both right. Explain how this is possible.

The two conditions are logically equivalent, so both can be the wp.

## 3 Proofs (30 points)

Do **either** proof A or proof B. For whichever one you choose:

a) Give appropriate loop invariants for any loops.

b) Write a **full** proof for the program. You may write a full proof outline or a Hilbert-style proof. (We would not recommend using a proof tree.)

If you start both, tell us which one you want us to grade: ___ Proof A ___ Proof B

### Proof A

This program calculates the square root of $n$ (if the square root isn't an integer, it returns the greatest integer less than the square root).

$$
\begin{array}{ll}
\textit{Precondition} & \{n \geq 0\} \\
 & i := \overline{1}; \\
 & \mathsf{while}\ i^2 \leq n\ \{ \\
\textit{Program} & \quad i := i + \overline{1} \\
 & \} \\
 & i := i - \overline{1} \\
\textit{Postcondition} & \{i^2 \leq n < (i+1)^2\}
\end{array}
$$

$$
\begin{array}{ll}
 & \{n \geq 0\} \Rightarrow \{0^2 \leq n\} \\
i := \overline{1}; & \{(i-1)^2 \leq n\} \\
\{\mathbf{inv}\ (i-1)^2 \leq n\} & \\
\mathsf{while}\ i^2 \leq n\ \{ & \{(i-1)^2 \leq n \wedge i^2 \leq n\} \Rightarrow \{i^2 \leq n\} \\
\quad i := i + \overline{1} & \{(i-1)^2 \leq n\} \\
\} & \{(i-1)^2 \leq n < i^2\} \\
i := i - \overline{1} & \{i^2 \leq n < (i+1)^2\}
\end{array}
$$

## Proof B

This program sets $r$ to true if all elements in $a$ are even, and false otherwise. "mod" is the modulus operator (% in C), and $x \bmod 2 = 0$ if and only if $x$ is even.

*Precondition* $\{T\}$

*Program*

$r :=$ true;<br>
$i := \overline{0}$;<br>
while $i < \mathit{size}(a)$ {<br>
  if $a[i] \bmod \overline{2} = \overline{0}$ then {<br>
    skip<br>
  } else {<br>
    $r :=$ false<br>
  };<br>
  $i := i + \overline{1}$<br>
}

*Postcondition* $\{r \leftrightarrow (\forall j \in [0, |a| - 1].a[j] \bmod 2 = 0)\}$

| | |
|---|---|
| | $\{T\} \Rightarrow \{(\forall j \in [0, -1].a[j] \bmod 2 = 0) \land 0 \leq \lvert a\rvert\}$ |
| $r :=$ true; | $\{r \leftrightarrow (\forall j \in [0, -1].a[j] \bmod 2 = 0) \land 0 \leq \lvert a\rvert\}$ |
| $i := \overline{0}$; | $\{r \leftrightarrow (\forall j \in [0, i-1].a[j] \bmod 2 = 0) \land i \leq \lvert a\rvert\}$ |
| $\{$**inv** $r \leftrightarrow (\forall j \in [0, i-1].a[j] \bmod 2 = 0) \land i \leq \lvert a\rvert\}$ | |
| while $i < \mathit{size}(a)$ { | |
| if $a[i] \bmod 2 = 0$ then { | $\{r \leftrightarrow (\forall j \in [0, i-1].a[j] \bmod 2 = 0) \land i + 1 \leq \lvert a\rvert \land a[i]$ mo |
| skip | $\{r \leftrightarrow (\forall j \in [0, i].a[j] \bmod 2 = 0) \land i + 1 \leq \lvert a\rvert\}$ |
| } else { | $\{r \leftrightarrow (\forall j \in [0, i-1].a[j] \bmod 2 = 0) \land i + 1 \leq \lvert a\rvert \land a[i]$ mo |
| $r :=$ false | $\{r \leftrightarrow (\forall j \in [0, i].a[j] \bmod 2 = 0) \land i + 1 \leq \lvert a\rvert\}$ |
| }; | $\{r \leftrightarrow (\forall j \in [0, i].a[j] \bmod 2 = 0) \land i + 1 \leq \lvert a\rvert\}$ |
| $i := i + \overline{1}$ | $\{r \leftrightarrow (\forall j \in [0, i-1].a[j] \bmod 2 = 0) \land i \leq \lvert a\rvert\}$ |
| } | $\{r \leftrightarrow (\forall j \in [0, i-1].a[j] \bmod 2 = 0) \land i \leq \lvert a\rvert \land i \geq \lvert a\rvert\} \Rightarrow \{$ |

## 4 Weakest Preconditions and Strongest Postconditions (24 points)

Calculate the following. Be sure to note whether we're asking for wp, wlp or sp. For full credit, **show your work** (this will also let us potentially give you partial credit if your final answer is wrong). You **do not need to** simplify your answers further once you have calculated a valid wp, wlp or sp.

(a) (6 points) $wlp(\textsf{if } x < \overline{0} \textsf{ then } \{x := x + \overline{3}\} \textsf{ else } \{y := y - \overline{3}\}, x < y)$

$$\begin{aligned} & (x < 0 \rightarrow wlp(x := x + \overline{3}, x < y)) \wedge (x \geq 0 \rightarrow wlp(y := y - \overline{3}, x < y)) \\ = \quad & (x < 0 \rightarrow x + 3 < y) \wedge (x \geq 0 \rightarrow x < y - 3) \end{aligned}$$

$wlp(\textsf{if } y > z \textsf{ then } \{x := x + \overline{3}\} \textsf{ else } \{y := y - \overline{3}\}, x < y)$

$$\begin{aligned} & (y > z \rightarrow wlp(x := x + \overline{3}, x < y)) \wedge (y \leq z \rightarrow wlp(y := y - \overline{3}, x < y)) \\ = \quad & (y > z \rightarrow x + 3 < y) \wedge (y \leq z \rightarrow x < y - 3) \end{aligned}$$

(b) (6 points) $wlp(j := i + \overline{2}, (\forall i. \overline{0} \leq i < j \rightarrow a[i] > j))$

$(\forall z. 0 \leq z < i + 2 \rightarrow a[z] > i + 2)$

$wlp(j := \overline{3} * i, (\forall i. \overline{0} \leq i < j \rightarrow a[i] = b[j]))$

$(\forall z. 0 \leq z < 3 * i \rightarrow a[z] = b[3 * i])$

(c) (6 points) $wp(j := x/y, \forall j. x \neq 2j)$

$$\begin{array}{ll} & [x/y/j](\forall j. x \neq 2j) \wedge D(j := x/y) \\ = & (\forall j. x \neq 2j) \wedge D(x/y) \\ = & (\forall j. x \neq 2j) \wedge y \neq 0 \end{array}$$

(d) (6 points) $sp(0 \leq x < y \wedge x = x_0 \wedge y = y_0, \textsf{if } y = \overline{5} \textsf{ then } \{x := x * \overline{2}\} \textsf{ else } \{y := y * \overline{2}\})$

$$
\begin{array}{ll}
& sp(0 \leq x < y \wedge y = 5 \wedge x = x_0 \wedge y = y_0, x := x * \overline{2}) \\
& \wedge sp(0 \leq x < y \wedge y \neq 5 \wedge x = x_0 \wedge y = y_0, y := y * \overline{2}) \\
= & (0 \leq x_0 < y \wedge y = 5 \wedge x = 2x_0 \wedge y = y_0) \\
& \wedge(0 \leq x < y_0 \wedge y_0 \neq 5 \wedge y = 2y_0 \wedge x = x_0)
\end{array}
$$

$sp(0 \leq x < y \wedge x = x_0 \wedge y = y_0, \textsf{if } x = \overline{2} \textsf{ then } \{x := x * \overline{5}\} \textsf{ else } \{y := y * \overline{5}\})$

$$
\begin{array}{ll}
& sp(0 \leq x < y \wedge x = 2 \wedge x = x_0 \wedge y = y_0, x := x * \overline{5}) \\
& \wedge sp(0 \leq x < y \wedge x \neq 2 \wedge x = x_0 \wedge y = y_0, y := y * \overline{5}) \\
= & (0 \leq x_0 < y \wedge x_0 = 2 \wedge x = 5x_0 \wedge y = y_0) \\
& \wedge(0 \leq x < y_0 \wedge x \neq 2 \wedge y = 5y_0 \wedge x = x_0)
\end{array}
$$

## 5 Proof Rules and Trees (14 points)

(a) (4 points) Your friend suggests that we could just use the following proof rule for assignments:

$$\frac{}{\vdash \{p\}\ x := e\ \{[e/x]p\}}$$

But, this rule is wrong. Give an *invalid* Hoare triple that could be proven using this rule.

$\{x > 5\}\ x := 0\ \{0 > 5\}$

(b) (10 points) Prove the following Hoare triple by writing a **proof tree**.

$$\vdash \{T\}\ x := y; x := x - 1\ \{x < y\}$$

$$\dfrac{\dfrac{\dfrac{}{\vdash \{y-1<y\}\ x := y\ \{x-1<y\}}\ (\textsc{AssignBwd}) \qquad \dfrac{}{\vdash \{x-1<y\}\ x := x-1\ \{x<y\}}\ (\textsc{AssignBwd})}{\vdash \{y-1<y\}\ x := y; x := x-1\ \{x<y\}}\ (\textsc{Seq}) \qquad T \Rightarrow y-1<y}{\vdash \{T\}\ x := y; x := x-1\ \{x<y\}}\ (\textsc{Weakening})$$

Prove the following Hoare triple by writing a **proof tree**.

$$\vdash \{T\}\ x := y; x := x + 1\ \{x > y\}$$

$$\dfrac{\dfrac{\dfrac{}{\vdash \{y+1>y\}\ x := y\ \{x+1>y\}}\ (\textsc{AssignBwd}) \qquad \dfrac{}{\vdash \{x+1>y\}\ x := x+1\ \{x>y\}}\ (\textsc{AssignBwd})}{\vdash \{y+1>y\}\ x := y; x := x+1\ \{x>y\}}\ (\textsc{Seq}) \qquad T \Rightarrow y+1>y}{\vdash \{T\}\ x := y; x := x+1\ \{x>y\}}\ (\textsc{Weakening})$$

# A Logic Laws

| Name | Description |
|---|---|
| Simplify | $p \wedge q \Rightarrow p, q$ |
| Modus Ponens | $(p \rightarrow q), p \Rightarrow q$ |
| Conjunction | $p, q \Rightarrow p \wedge q$ |
| Disjunction | $p \Rightarrow p \vee q, q \vee p$ |
| Definition of Conditional | $p \rightarrow q \Leftrightarrow \neg p \vee q$ |
| Definition of Biconditional | $p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$ |
| Law of the Excluded Middle (LEM) | $p \vee \neg p \Leftrightarrow T$ |
| Double Negation Elimination (DNE) | $p \Leftrightarrow \neg\neg p$ |
| Contradiction | $p \wedge \neg p \Leftrightarrow F$ |
| Identity | $p \wedge T \Rightarrow p$, $p \vee F \Rightarrow p$ |
| DeMorgan's Laws | $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$<br>$\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$<br>$\neg(\forall x.p(x)) \Leftrightarrow \exists x.\neg p(x)$<br>$\neg(\exists x.p(x)) \Leftrightarrow \forall x.\neg p(x)$ |
| Distributivity | $(p \wedge q) \vee r \Leftrightarrow (p \vee r) \wedge (q \vee r)$<br>$(p \vee q) \wedge r \Leftrightarrow (p \wedge r) \vee (q \wedge r)$ |
| Commutativity | $p \wedge q \Leftrightarrow q \wedge p$, $p \vee q \Leftrightarrow q \vee p$ |
| Associativity | $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$, $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$ |
| Idempotency | $p \wedge p \Leftrightarrow p$, $p \vee p \Leftrightarrow p$ |
| Domination | $p \vee T \Leftrightarrow T$, $p \wedge F \Leftrightarrow F$ |

## B Language Syntax and Semantics

### Expression and Statement Syntax

$$
\begin{array}{rcl}
e & ::= & \overline{n} \mid \mathsf{true} \mid \mathsf{false} \mid x \mid a[e] \mid e\ op\ e \mid e\ ?\ e : e \mid size(a) \\
s & ::= & \mathsf{skip} \mid s; s \mid x := e \mid a[e] := e \mid \mathsf{if}\ e\ \mathsf{then}\ \{s\}\ \mathsf{else}\ \{s\} \mid \mathsf{while}\ e\ \{s\}
\end{array}
$$

### Expression Semantics

$$
\begin{array}{rcll}
\sigma(\overline{n}) & = & n & \\
\sigma(\mathsf{true}) & = & T & \\
\sigma(\mathsf{false}) & = & F & \\
\sigma(x) & = & \sigma(x) & \\
\sigma(a[e]) & = & (\sigma(a))[\sigma(e)] & \sigma(e) \neq \bot_e \wedge 0 \leq \sigma(e) < |\sigma(a)| \\
\sigma(a[e]) & = & \bot_e & \text{otherwise} \\
\sigma(e_1\ op\ e_2) & = & \sigma(e_1)\ op\ \sigma(e_2) & \sigma(e_1) \neq \bot_e \neq \sigma(e_2) \\
\sigma(e_1\ op\ e_2) & = & \bot_e & \sigma(e_1) = \bot_e \vee \sigma(e_2) = \bot_e \\
\sigma(e_1\ ?\ e_2 : e_3) & = & \sigma(e_2) & \sigma(e_1) = T \\
\sigma(e_1\ ?\ e_2 : e_3) & = & \sigma(e_3) & \sigma(e_1) = F \\
\sigma(e_1\ ?\ e_2 : e_3) & = & \bot_e & \sigma(e_1) = \bot_e \\
\sigma(size(a)) & = & |\sigma(a)| &
\end{array}
$$

### Statement Semantics - Small-step

$$\frac{\langle s_1, \sigma\rangle \to \langle s_1', \sigma\rangle}{\langle s_1; s_2, \sigma\rangle \to \langle s_1'; s_2, \sigma\rangle} \qquad \frac{\langle s_1, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle}{\langle s_1; s_2, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle} \qquad \frac{}{\langle \mathsf{skip}; s, \sigma\rangle \to \langle s, \sigma\rangle}$$

$$\frac{\sigma(e) \neq \bot_e}{\langle x := e, \sigma\rangle \to \langle \mathsf{skip}, \sigma[x \mapsto \sigma(e)]\rangle} \qquad \frac{\sigma(e) = \bot_e}{\langle x := e, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle}$$

$$\frac{\sigma(e_1) \neq \bot_e \qquad \sigma(e_2) \neq \bot_e \qquad 0 \leq \sigma(e_1) < |\sigma(a)|}{\langle a[e_1] := e_2, \sigma\rangle \to \langle \mathsf{skip}, \sigma[a[\sigma(e_1)] \mapsto \sigma(e_2)]\rangle} \qquad \frac{\sigma(e_1) = \bot_e \vee \sigma(e_2) = \bot_e}{\langle a[e_1] := e_2, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle}$$

$$\frac{\sigma(e_1) \geq |\sigma(a)| \vee \sigma(e_1) < 0}{\langle a[e_1] := e_2, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle} \qquad \frac{\sigma(e) = T}{\langle \mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}, \sigma\rangle \to \langle s_1, \sigma\rangle}$$

$$\frac{\sigma(e) = F}{\langle \mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}, \sigma\rangle \to \langle s_2, \sigma\rangle} \qquad \frac{\sigma(e) = \bot_e}{\langle \mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}, \sigma\rangle \to \langle \mathsf{skip}, \bot_e\rangle}$$

$$\frac{}{\langle \mathsf{while}\ e\ \{s\}, \sigma\rangle \to \langle \mathsf{if}\ e\ \mathsf{then}\ \{s; \mathsf{while}\ e\ \{s\}\}\ \mathsf{else}\ \{\mathsf{skip}\}, \sigma\rangle}$$

**Statement Semantics - Big-step**

$$\begin{array}{rcll}
M(\mathsf{skip}, \sigma) & = & \{\sigma\} & \\
M(s_1; s_2, \sigma) & = & \bigcup_{\sigma' \in M(s_1,\sigma)} M(s_2, \sigma') & \\
M(x := e, \sigma) & = & \{\sigma[x \mapsto \sigma(e)]\} & \sigma(e) \neq \bot_e \\
M(x := e, \sigma) & = & \{\bot_e\} & \sigma(e) = \bot_e \\
M(a[e_1] := e_2, \sigma) & = & \{\sigma[a[\sigma(e_1)] \mapsto \sigma(e_2)]\} & \sigma(e_1) \neq \bot_e \wedge \sigma(e_2) \neq \bot_e \wedge 0 \leq \sigma(e_1) < |\sigma(a)| \\
M(a[e_1] := e_2, \sigma) & = & \{\bot_e\} & \text{otherwise} \\
M(\mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}, \sigma) & = & M(s_1, \sigma) & \sigma(e) = T \\
M(\mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}, \sigma) & = & M(s_2, \sigma) & \sigma(e) = F \\
M(\mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}, \sigma) & = & \{\bot_e\} & \sigma(e) = \bot_e \\
M(\mathsf{while}\ e\ \{s\}, \sigma) & = & \Sigma_k & \Sigma_k \text{ is the lowest } k \text{ such that if} \\
 & & & \sigma \in \Sigma_k \text{ , then } \sigma(e) = F \\
M(\mathsf{while}\ e\ \{s\}, \sigma) & = & \{\bot_d\} & \text{no such } k \text{ exists}
\end{array}$$

where

$$\begin{array}{rcl}
\Sigma_0 & = & \{\sigma\} \\
\Sigma_k + 1 & = & \bigcup_{\sigma \in \Sigma_k} M(s, \sigma)
\end{array}$$

# C Hoare Triple Inference Rules

$$\frac{}{\vdash \{p\}\ \mathsf{skip}\ \{p\}}\ (\textsc{Skip}) \qquad \frac{}{\vdash \{[e/x]p\}\ x := e\ \{p\}}\ (\textsc{AssignBwd})$$

$$\frac{}{\vdash \{[e/a[i]]p\}\ a[i] := e\ \{p\}}\ (\textsc{ArrAssignBwd})$$

$$\frac{x_0\ \mathsf{fresh}}{\vdash \{p\}\ x := e\ \{[x_0/x]p \wedge x = [x_0/x]e\}}\ (\textsc{AssignFwd}) \qquad \frac{\vdash \{p\}\ s_1\ \{q'\} \qquad \vdash \{q'\}\ s_2\ \{q\}}{\vdash \{p\}\ s_1; s_2\ \{q\}}\ (\textsc{Seq})$$

$$\frac{\vdash \{p \wedge e\}\ s_1\ \{q_1\} \qquad \vdash \{p \wedge \neg e\}\ s_2\ \{q_2\}}{\vdash \{p\}\ \mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}\ \{q_1 \vee q_2\}}\ (\textsc{If}) \qquad \frac{\vdash \{p \wedge e\}\ s_1\ \{q\} \qquad \vdash \{p \wedge \neg e\}\ s_2\ \{q\}}{\vdash \{p\}\ \mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}\ \{q\}}\ (\textsc{If'})$$

$$\frac{\vdash \{p \wedge e\}\ s\ \{p\}}{\vdash \{p\}\ \mathsf{while}\ e\ \{s\}\ \{p \wedge \neg e\}}\ (\textsc{While}) \qquad \frac{\vdash \{p\}\ s\ \{q\} \qquad p' \Rightarrow p \qquad q \Rightarrow q'}{\vdash \{p'\}\ s\ \{q'\}}\ (\textsc{Weaken})$$

## D Calculating wp, wlp, sp for loop-free programs

$\oplus$ stands for any binary operator that doesn't itself cause errors, e.g., +, -...

$$wlp(\mathsf{skip}, q) = q \qquad wlp(x := e, q) = [e/x]q$$
$$wlp(a[e_1] := e_2, q) = [e_2/a[e_1]]q \qquad wlp(s_1; s_2, q) = wlp(s_1, wlp(s_2, q))$$

$$wlp(\mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}, q) = (e \to wlp(s_1, q)) \wedge (\neg e \to wlp(s_2, q))$$

$$sp(p, \mathsf{skip}) = p \qquad sp(p, x := e) = [x_0/x]p \wedge x = [x_0/x]e$$
$$sp(p, s_1; s_2) = sp(sp(p, s_1), s_2)$$

$$sp(p, \mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}) = sp(p \wedge e, s_1) \vee sp(p \wedge \neg e, s_2)$$

$$D(c) = T \qquad D(x) = T$$
$$D(a[e]) = D(e) \wedge 0 \le e < |a| \qquad D(e_1/e_2) = D(e_1) \wedge D(e_2) \wedge e_2 \neq 0$$
$$D(sqrt(e)) = D(e) \wedge e \ge 0 \qquad D(e_1 \oplus e_2) = D(e_1) \wedge D(e_2)$$

$$D(e_1\ ?\ e_2 : e_3) = D(e_1) \wedge (e_1 \to D(e_2)) \wedge (\neg e_1 \to D(e_3))$$

$$D(\mathsf{skip}) = T \qquad D(x := e) = D(e)$$
$$D(a[e_1] := e_2) = D(a[e_1]) \wedge D(e_2) \qquad D(s_1; s_2) = D(s_1) \wedge wp(s_1, D(s_2))$$
$$D(\mathsf{while}\ e\ \{s\}) = D(e) \wedge (e \to D(s))$$

$$D(\mathsf{if}\ e\ \mathsf{then}\ \{s_1\}\ \mathsf{else}\ \{s_2\}) = D(e) \wedge (e \to D(s_1)) \wedge (\neg e \to D(s_2))$$

$$wp(s, q) = wlp(s, q) \wedge D(s)$$

## E Simplifying Conditional Expressions

$$T\ ?\ e_1 : e_2 \Rrightarrow e_1 \quad \text{Always} \qquad F\ ?\ e_1 : e_2 \Rrightarrow e_2 \quad \text{Always}$$
$$e_0\ ?\ e : e \Rrightarrow e \quad \text{Always}$$
$$e_0\ ?\ e_1 : e_2 \Rrightarrow e_2 \quad \text{If } e_0 \Rightarrow e_1 = e_2 \qquad e_0\ ?\ e_1 : e_2 \Rrightarrow e_1 \quad \text{If } \neg e_0 \Rightarrow e_1 = e_2$$

Let $\Theta$ be a unary operator, $\oplus$ be a binary operator or relation and $f$ be any function.

$$\Theta(e\ ?\ e_1 : e_2) \Rrightarrow e\ ?\ \Theta(e_1) : \Theta(e_2) \qquad (e\ ?\ e_1 : e_2) \oplus e_3 \Rrightarrow e\ ?\ e_1 \oplus e_3 : e_2 \oplus e_3$$
$$a[e\ ?\ e_1 : e_2] \Rrightarrow e\ ?\ a[e_1] : a[e_2] \qquad f(e\ ?\ e_1 : e_2) \Rrightarrow e\ ?\ f(e_1) : f(e_2)$$

If $e, e_1$, and $e_2$ are Boolean expressions, then

$$(e\ ?\ e_1 : F) \Leftrightarrow (e \wedge e_1) \qquad (e\ ?\ F : e_2) \Leftrightarrow (\neg e \wedge e_2)$$
$$(e\ ?\ e_1 : T) \Leftrightarrow (e \to e_1) \Leftrightarrow (\neg e \vee e_1) \qquad (e\ ?\ T : e_2) \Leftrightarrow (\neg e \to e_2) \Leftrightarrow (e \vee e_2)$$

$$(e\ ?\ e_1 : e_2) \Leftrightarrow ((e \to e_1) \wedge (\neg e \to e_2)) \Leftrightarrow ((e \wedge e_1) \vee (\neg e \wedge e_2))$$