CS536, Spring 2022
Practice Midterm Exam #1
SOLUTIONS: DO NOT DISTRIBUTE

Name ______________________________

IIT Email ______________________________

Important notes:

- 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.
- The last 3 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 one double-sided 8.5" $\times$ 11" sheet of notes. We will collect your sheet of notes at the end of the exam, so if you want it back, please make sure your name is on it. **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____________________

1 True and False (20 points)

1. F Careful testing can find any bug.

2. T P is a tautology if and only if $T \Rightarrow P$

3. T $\{T\}\ s\ \{T\}$ is valid for any s.

4. F $[\ T\ ]\ s\ [\ T\ ]$ is valid for any s.

5. T For any p and q, $(p \land \neg p) \Rightarrow q$.

6. T $\vDash \exists x \in \mathbb{Z}. x > y$ is valid (satisfied in all states).

7. F If $\{p\}\ s\ \{q\}$ is valid, then q must always be true after running s, no matter what.

8. T If s doesn't diverge or error, then $\vDash \{p\}\ s\ \{q\}$ if and only if $\vDash [\ p\ ]\ s\ [\ q\ ]$.

9. F $M(\mathsf{if}\ x > 0\ \mathsf{then}\ \{y := 0\}\ \mathsf{else}\ \{y := 1\}, \sigma)$ is a set containing two states, for any σ.

10. F $\{x = 1, y = 1\} \vDash \forall x \in \mathbb{Z}. x = y$

2 Make a Predicate (6 points)

Give the definition of a predicate function $P(a, b)$ that is true iff every element of the array a is greater than ($>$) some element of the array b. For example, $P(a, b)$ should be satisfied in the state $\{a = [2; 5; 6], b = [8; 1; 3; 9]\}$ but unsatisfied in $\{a = [1; 5; 7], b = [8; 1; 3; 9]\}$ (because there is no element in b strictly less than 1). You can use, e.g., $|a|$ to refer to the size of a in your predicate. Recall that arrays are 0-indexed.

$\forall i \in \mathbb{Z}.(i \geq 0 \land i < |a|) \to \exists j \in \mathbb{Z}.j \geq 0 \land j < |b| \land a[i] > b[j]$

3 Truth Table (12 points)

(a) Construct a truth table for the proposition

$$(P \vee Q) \wedge (\neg P \vee Q)$$

P	Q	$\neg P$	$(P \vee Q)$	$(\neg P \vee Q)$	$(P \vee Q) \wedge (\neg P \vee Q)$
T	T	F	T	T	T
T	F	F	T	F	F
F	T	T	T	T	T
F	F	T	F	T	F

(b) Is $(P \rightarrow Q) \leftrightarrow (Q \rightarrow P)$ a tautology, contradiction or contingency?

Contingency

4 Proof (22 points)

Prove the following logical implication (on HW1, you saw *uncurrying*; this is *currying*).

$$((P \land Q) \to R) \Rightarrow (P \to (Q \to R))$$

Write a statement and a justification on each line, as you did on HW1. The justifications should be logical laws (See Appendix A) with a reference to the lines you're using. For example, if line 3 has the statement P and line 4 has the statement Q, you could justify $P \land Q$ with "Conjunction(3, 4)". If a justification uses only one line and it's the line immediately before, you can leave it out. You can leave lines left over; there may be more than you need.

	Statement	Justification
1	$(P \land Q) \to R$	Assumption
2	$\neg(P \land Q) \lor R$	Definition of Conditional
3	$(\neg P \lor \neg Q) \lor R$	DeMorgan
4	$\neg P \lor (\neg Q \lor R)$	Associativity
5	$\neg P \lor (Q \to R)$	Definition of Conditional
6	$P \to (Q \to R)$	Definition of Conditional

5 Hoare Triples (20 points)

(a) (15 points) For each triple below, say whether or not it's satisfied in the given state and explain briefly.

i) $\{x = 0\} \models \{x > 0\}\ x := x - \overline{1}\ \{x > 0\}$
Yes. Precondition is unsatisfied.

ii) $\{x = 5\} \models \{x > 0\}\ x := x - \overline{1}\ \{x > 0\}$
Yes. Precondition is satisfied, and final $x > 0$

iii) $\{x = 0\} \models [\ T\ ]\ \textsf{while}\ x \neq \overline{0}\ \{\textsf{skip}\}\ [\ x = 0\ ]$
Yes. Program terminates with $x = 0$.

iv) $\{x = 1\} \models [\ T\ ]\ \textsf{while}\ x \neq \overline{0}\ \{\textsf{skip}\}\ [\ x = 0\ ]$
No. Program fails to terminate.

v) $\{x = 1\} \models \{T\}\ \textsf{while}\ x \neq \overline{0}\ \{\textsf{skip}\}\ \{x = 0\}$
Yes. Program fails to terminate.

(b) (5 points) Write a program, using the syntax of our language from class, that satisfies the pre- and post-conditions below.

$\models [\ n \geq 0 \wedge n < |a|\ ]$ ________ $[\ r = a[|a|] \times a[|a| - 1] \times \cdots \times a[|a| - n]\ ]$

```
r := 1;
i := 0;
while(i <= n) {
    r := r * a[size(a) - i];
    i := i + 1
}
```

6 Take some steps (20 points)

Take the following program, which we will call s.

$$
\begin{array}{l}
i := \overline{0}; \\
\mathsf{while}(a[i] > \overline{0})\{ \\
\quad i := a[i] \\
\}
\end{array}
$$

Let $\sigma = \{a = [2, -1, 3]\}$. Recall that arrays are *0-indexed* (so, e.g., $\sigma(a[0]) = 2$).

(a) (16 points) Fill in the blanks in the following small-step evaluation of s. If you reach skip before you run out of lines, cross out the remaining lines. When writing states, write them as sets of variables and values, e.g. $\{x = 1, y = 2\}$, not as state updates (e.g. $\{x = 2, y = 1\}[x \mapsto 1]$).

$\langle s, \{a = [2, -1, 3]\}\rangle$

$\rightarrow^2$ $\langle \mathsf{while}\ a[i] > \overline{0}\ \{i := a[i]\}, \{a = [2, 1, 3], i = 0\}\rangle$

$\rightarrow^2$ $\langle i := a[i]; \mathsf{while}\ a[i] > \overline{0}\ \{i := a[i]\}, \{a = [2, 1, 3], i = 0\}\rangle$ because $\sigma(a[i] > \overline{0}) = T$

$\rightarrow^2$ $\langle \mathsf{while}\ a[i] > \overline{0}\ \{i := a[i]\}, \{a = [2, 1, 3], i = 2\}\rangle$

$\rightarrow^2$ $\langle i := a[i]; \mathsf{while}\ a[i] > \overline{0}\ \{i := a[i]\}, \{a = [2, 1, 3], i = 2\}\rangle$ because $\sigma(a[i] > \overline{0}) = T$

$\rightarrow^2$ $\langle \mathsf{while}\ a[i] > \overline{0}\ \{i := a[i]\}, \{a = [2, 1, 3], i = 3\}\rangle$

$\rightarrow^2$ $\langle \mathsf{skip}, \{\perp_e\}\rangle$ because $\sigma(a[i] > \overline{0}) = \perp_e$

$\rightarrow^2$ $\langle \mathsf{while}\ a[i] > \overline{0}\ \{i := a[i]\}, X\rangle$

$\rightarrow^2$ $\langle X, X\rangle$ because $\sigma(a[i] > \overline{0}) = X$

[QUESTION CONTINUES ON NEXT PAGE]

(b) (4 points) Use the big-step semantics to figure out the final state we'll reach by running s in each of the initial states below. Use $\perp_d$ for divergence and $\perp_e$ for errors; do not use just $\perp$ (with no subscript). You don't need to show work. Note that the initial states are different from above.

i) $M(s, \{a = [1, -1]\})$ $\{a = [1, -1], i = 1\}$

ii) $M(s, \{a = [1, 2, 1]\})$ $\{\perp_d\}$

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	$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$ $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$ $\neg(\forall x.p(x)) \Leftrightarrow \exists x.\neg p(x)$ $\neg(\exists x.p(x)) \Leftrightarrow \forall x.\neg p(x)$
Distributivity	$(p \wedge q) \vee r \Leftrightarrow (p \vee r) \wedge (q \vee r)$ $(p \vee q) \wedge r \Leftrightarrow (p \wedge r) \vee (q \wedge r)$
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

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