Final Exam
CSE6049 Program Analysis, Spring 2021
6/21(Mon), 16 : 00

Name: **asdfasdfasdfasdfasdf**

Student ID: **asdfasdfasdfasdfasdfasdf**

Problem 1. [O/X questions] (20 pts). Mark O for each correct statement (X for wrong statement). You will get 2 points per correct answer, but you will lose 2 points for each wrong answer. Leave a blank when you are uncertain.

a) A binary relation is a partial order if it has reflexivity, antisymmetry, and transitivity (O, X) (O)

b) The powerset of integers $(\wp(\mathbb{Z}), \subseteq)$ is a CPO. (O, X) (O)

c) If a function is monotone, then it is also continuous. (O, X) (X)

d) Best abstraction is always obtainable. (O, X) (X)

e) Suppose $D \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \hat{D}$ for some CPOs D and $\hat{D}$. Both α and γ are monotone. (O, X) (O)

f) We can build a sound and complete static analyzer for any kinds of non-trivial properties. (O, X) (X)

g) Proving absence of invariant violations or crashing errors are examples of proving safety properties. (O, X) (O)

h) Fully context-sensitive analysis is cheap in general, thus feasible in practice. (O, X) (X)

i) We can express any kinds of static analysis in Datalog. (O, X) (X)

j) Sparse analysis improves performance of the original analysis by sacrificing precision. (O, X) (X).

Problem 2. [Spectrum of Program Analysis Techniques], (10 pts). Rice's theorem is as follows:

Let $\mathbb{L}$ be a Turing-complete language, and let P be a nontrivial semantic property of program of $\mathbb{L}$. There exists no *automatic* and *eventually terminating* method such that,

For *every* program p in $\mathbb{L}$, it returns true *if and only if* p satisfies the semantic property P.

Choose either one of

1. Machine-assisted proving
2. Finite-state model checking,
3. Testing
4. Domain-specific verifier

for each of the following cases of giving up something among the above keywords.

- "automatic": [1]
- "every": [4]
- "eventually terminating": [2]
- "if and only if": [3]

Problem 3. [Soundness & Completeness], (10 pts). What are the pros and cons of a program analyzer which is unsound but complete? What does this analyzer guarantee?

Pros: no false alarm.
Cons: cannot cover all errors.

Problem 4. [Pointer analysis], (10 pts). Write the result of flow- and context-insensitive pointer analysis of the following program.

```
f(v) {
  u = v;
  return u;
}
x = &h1;
z = &h2;
y = f(x);
w = f(z);
```

$$\{\texttt{x} \rightarrow \texttt{h1}, \texttt{z} \rightarrow \texttt{h2}, \boxed{\text{y} \rightarrow h1, y \rightarrow h2, w \rightarrow h1, w \rightarrow h2, u \rightarrow h1, u \rightarrow h2, v \rightarrow h1, v \rightarrow h2}\}$$

Problem 5. [Galois connection] (10 pts). The following is a Galois connection to abstract a set of integers into a set $\hat{Z}$ of their remainders modulo 4. For example, $\{14, 22\}$ can be abstracted to $\{2\}$.

$$\wp(\mathbb{Z}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \hat{\mathbb{Z}} = \wp(\{0, 1, 2, 3\})$$

Complete the definition of γ.

$$\begin{array}{rcl}
\alpha(\emptyset) & = & \emptyset \\
\alpha(X) & = & \{n \bmod 4 \mid n \in X\} \\
\gamma(\emptyset) & = & \emptyset \\
\gamma(\hat{X}) & = & \boxed{\{\text{ n} \mid (n \bmod 4) \in \hat{X}\}}
\end{array}$$

Problem 6. [Collecting semantics] (20 pts). Consider the following simple language:

E	$::=$	n	integer constants
	$\mid$	x	variable
	$\mid$	$E + E$	binary operation
B	$::=$	$x < E$	comparison expressions
	$\mid$	$\neg B$	negation expressions
C	$::=$	`skip`	skip
	$\mid$	C ; C	sequence
	$\mid$	$x := E$	assignment command
	$\mid$	`input`(x)	external input
	$\mid$	`if` B { C } `else` { C }	conditional command
	$\mid$	`while` B C	loop command

The collecting semantics can be described as *denotational semantics*:

$$\begin{array}{rcl} [\![C]\!] & \in & \wp(\mathbb{M}) \to \wp(\mathbb{M}) \\ [\![E]\!] & \in & \wp(\mathbb{M}) \to \wp(\mathbb{Z}) \\ [\![B]\!] & \in & \wp(\mathbb{M}) \to \wp(\mathbb{M}) \\ \mathbb{M} & = & \mathbb{X} \to \mathbb{Z} \end{array}$$

where $\wp(\mathbb{M})$ denotes the powerset of memories, $\mathbb{X}$ is the set of variables in a given program and $\mathbb{Z}$ is the set of integers. Define the collecting semantic functions by filling the holes in the followings.

$$\begin{array}{rcl} [\![n]\!](M) & = & \{n\} \\ [\![x]\!](M) & = & \{m(x) \mid m \in M\} \\ [\![E_1 + E_2]\!](M) & = & \boxed{\{ \text{ v}_1 + v_2 \mid v_1 \in [\![E_1]\!](M), v_2 \in [\![E_1]\!](M)\}} \end{array}$$

$$\begin{array}{rcl} [\![x < E]\!](M) & = & \{m \in M \mid m(x) < v, v \in [\![E]\!](\{m\})\} \\ [\![\neg B]\!](M) & = & M \setminus [\![B]\!](M) \end{array}$$

$$\begin{array}{rcl} [\![\texttt{skip}]\!](M) & = & M \\ [\![C_1\ ;\ C_2]\!](M) & = & \boxed{[\![\text{ C}_2]\!]([\![C_1]\!](M))} \\ [\![x := E]\!](M) & = & \{m[x \mapsto v] \mid v \in [\![E]\!](M), m \in M\} \\ [\![\texttt{input}(x)]\!](M) & = & \boxed{\{\text{ m[x} \mapsto v] \mid v \in \mathbb{Z}, m \in M\}} \\ [\![\texttt{if}\ B\ C_1\ \texttt{else}\ C_2]\!](M) & = & \boxed{[\![\text{ C}_1]\!]([\![B]\!](M)) \cup [\![C_2]\!]([\![\neg B]\!](M))} \\ [\![\texttt{while}\ B\ C]\!](M) & = & [\![\neg B]\!](\mathbf{lfp}_M F) \end{array}$$

where

$$F = \lambda X.\ \boxed{\text{M} \cup [\![C]\!]([\![B]\!](X))}$$

Problem 7. [Widening] (10pts). Write the conditions of widening operators ($\triangledown$) on an abstract domain $\mathbb{A}$.

1. $\forall a, b \in \mathbb{A}.\ a \sqsubseteq a \triangledown b \wedge b \sqsubseteq a \triangledown b$

2. For all sequence $(a_n)_{n\in\mathbb{N}}$ of abstract elements, the sequence $(a'_n)_{n\in\mathbb{N}}$ defined below is ultimately stationary:

$$\begin{array}{rcl} a'_0 & = & \boxed{\text{a}_0} \\ a'_{n+1} & = & \boxed{\text{a'}_n \triangledown a_n} \end{array}$$

Problem 8. [Fixpoint Transfer Theorem] (20pts). Complete a fraction of the following proof of the fixpoint transfer theorem which says:

Let $D \leftrightarrows D^{\#}$ where D and $D^{\#}$ are CPOs. If we have a continuous function $F : D \to D$ and a monotone function $F^{\#} : D^{\#} \to D^{\#}$ such that $F \circ \gamma \sqsubseteq \gamma \circ F^{\#}$. Then,

$$\mathbf{lfp}F \sqsubseteq \gamma(\bigsqcup_{i \in \mathbb{N}} F^{\#^i}(\bot^{\#})))$$

Proof. First we prove

$$\forall n \in \mathbb{N}.\ F^n(\bot) \sqsubseteq \gamma(F^{\#^n}(\bot^{\#})$$

by induction. The base case is trivial. The inductive case is to show that

$$F^n(\bot) \sqsubseteq \gamma(F^{\#^n}(\bot^{\#})) \implies F^{n+1}(\bot) \sqsubseteq \gamma(F^{\#^{n+1}}(\bot^{\#})).$$

which can be proven as follows:

$$\begin{array}{lcll}
F^{n+1}(\bot) & = & F \circ F^n(\bot) & \\
 & \sqsubseteq & F \circ \gamma(F^{\#^n}(\bot^{\#})) & \text{(because } \boxed{\text{by induction hypothesis and monotonicity of F}}\text{)} \\
 & \sqsubseteq & \gamma \circ F^{\#} \circ F^{\#^n}(\bot^{\#}) & \text{(because } \boxed{\text{by assumption F } \circ\gamma \sqsubseteq \gamma \circ F^{\#}}\text{)} \\
 & = & \gamma(F^{\#^{n+1}}(\bot^{\#})) &
\end{array}$$

Therefore,

$$\begin{array}{lcll}
\mathbf{lfp}F & = & \bigsqcup_{i \geq 0} F^i(\bot) & \\
 & \sqsubseteq & \bigsqcup_{i \geq 0} \gamma(F^{\#^i}(\bot^{\#})) & \\
 & \sqsubseteq & \gamma(\bigsqcup_{i \geq 0}(F^{\#^i}(\bot^{\#}))) & \text{(by monotonicity of } \gamma\text{)}
\end{array}$$

□

Problem 9. [Safe Memory Access] (10 pts). Suppose we analyze the following program based on the interval domain. What will be the most precise interval values we can compute for variables `x`, `y`, and `z` at the end of the program?

```
x = 0;
y = 2;
if (*) { p = &x; }
else { p = &y; }
z = *p;
*p = 1;
```

- `x`: [0, 1]
- `y`: [1, 2]
- `z`: [0, 2]