Practice Exam

15-414/614 Bug Catching: Automated Program Verification

Name: _____

Andrew ID: _____

Instructions

- This exam is closed-book.
- You have XX minutes to complete the exam.
- There are 7 problems on 12 pages.
- Read each problem carefully before attempting to solve it.
- State any assumptions that you make about a question.
- If you aren't sure about an assumption, ask the course staff.

	Max	Score
Dynamic Logic	40	
Resolution	30	
SAT Solvers	30	
SAT Encodings	20	
Arrays and Uninterpreted Functions	25	
Certificates	30	
Temporal Logic	30	
Total:	205	

1 Dynamic Logic (40 points)

This problem explores an alternative application of dynamic logic. Instead of reasoning about imperative programs, we reason about programs for a simple stack machine.

Consider the following set of *programs* α, where we replace the usual assignment with several stack operations.

$$\begin{array}{llll} \text{Programs} & \alpha, \beta & ::= & \mathbf{push}\ k \mid \mathbf{dup} \mid \mathbf{drop} \mid \mathbf{dec} \mid \mathbf{plus} \mid \mathbf{times} \\ & & & \mid \alpha\ ;\ \beta \mid \alpha \cup \beta \mid ?P \mid \alpha^* \\ \text{States} & s, t & ::= & k_1 \cdots k_n \\ \text{Formulas} & P & ::= & \mathbf{top}\ k \mid \mathbf{true} \mid \mathbf{false} \\ & & & \neg P \mid P \wedge Q \mid P \to Q \mid \forall x.\, P \mid \exists x.\, P \mid P \vee Q \mid [\alpha]P \mid \langle\alpha\rangle P \end{array}$$

States are just *stacks* of integers $k_1 \cdots k_n$ where k_1 is the top of the stack. *Formulas* no longer mention variables (which the language of programs does not have). Instead we have a single new formula $\mathbf{top}\ k$ which holds if the top of the current stack is equal to the number k. The quantifiers here range over integers, as usual, and we imagine we state additional arithmetic properties.

We give the semantic definitions for the new constructs; all the other cases remain the same.

$$\begin{array}{lll} s[\![\mathbf{push}\ k]\!]s' & \text{iff} & s' = k \cdot s \\ s[\![\mathbf{drop}]\!]s' & \text{iff} & s = k \cdot s' \quad \text{for some } k \\ s[\![\mathbf{dup}]\!]s' & \text{iff} & s = k \cdot t \text{ and } s' = k \cdot k \cdot t \quad \text{for some } k \text{ and } t \\ s[\![\mathbf{dec}]\!]s' & \text{iff} & s = k \cdot t \text{ and } s' = (k-1) \cdot t \quad \text{for some } k \text{ and } t \\ s[\![\mathbf{minus}]\!]s' & \text{iff} & s = k_1 \cdot k_2 \cdot t \text{ and } s' = (k_1 - k_2) \cdot t \quad \text{for some } k_1, k_2, \text{ and } t \\ s[\![\mathbf{times}]\!]s' & \text{iff} & s = k_1 \cdot k_2 \cdot t \text{ and } s' = (k_1 \times k_2) \cdot t \quad \text{for some } k_1, k_2, \text{ and } t \\ s \models \mathbf{top}\ k & \text{iff} & s = k \cdot t \quad \text{for some } t \end{array}$$

For example, for any stack s we will have

$$s[\![\mathbf{push}\ k\ ;\ \mathbf{times}]\!](k \times k) \cdot s$$

10 **Task 1** Describe the meaning of the following program as a relation between an empty initial stack and a final stack s' by stating the possible forms of s'.

$(\cdot)[\![\mathbf{push}\ k\ ;\ \mathbf{push}\ i\ ;\ \mathbf{push}\ j\ ;\ \mathbf{minus}\ ;\ \mathbf{times}]\!]s' \quad \text{iff}$

10 **Task 2** Describe the meaning of the following program as a relation between an initial stack just containing $n > 0$ and a final stack s', by stating the possible forms of s'.

$n[\![(?\neg(\mathbf{top}\ 1)\ ;\ \mathbf{dup}\ ;\ \mathbf{dec})^*\ ;\ ?\mathbf{top}\ 1\ ;\ \mathbf{times}^*]\!]s'$ iff

20 **Task 3** Let f be a mathematical function from an integer to an integer. Write a formula $\mathsf{computes}\ f\ \alpha$ such that for every integer k and stack s we have $k \models \mathsf{computes}\ f\ \alpha$ iff $k[\![\alpha]\!]f(k) \cdot s'$ for some s'. Your formula may mention f applied to an argument.

$\mathsf{computes}\ f\ \alpha =$

Prove the correctness of your definition

2 Resolution (30 points)

15 **Task 1** A *tautology* is a clause that contains an atom p and also its negation $\neg p$. Let $\mathcal{T}$ be a set of propositional clauses. Prove that if we delete all tautologies from $\mathcal{T}$ to obtain $\mathcal{S}$, then $\mathcal{T}$ and $\mathcal{S}$ have the same set of satisfying assignments.

15 **Task 2** We say clause C *subsumes* clause D if $C \subseteq D$ and *strictly subsumes* clause D if $C \subsetneq D$. Let $\mathcal{T}$ be a set of propositional clauses. Let $\mathcal{S}$ be the result of deleting all clauses from $\mathcal{T}$ that are strictly subsumed by other clauses in $\mathcal{T}$. Prove that $\mathcal{T}$ and $\mathcal{S}$ have the same set of satisfying assignments.

3 SAT Solvers (30 points)

20 **Task 1** DPLL learns new clauses that help it avoid entering conflicts similar to those it has already encountered. Consider the following alternative method, which is easier to implement.

1. Let $(l_1, \ldots, l_n)$ be a partial assignment that results in a conflict.
2. Add the clause $\neg l_1 \vee \neg l_2 \vee \cdots \vee \neg l_n$ to the set of clauses as a learned clause.

(10 points) Is this approach sound, i.e. will adding these clauses potentially change the satisfiability of the original formula? Justify your answer.

(10 points) Is this approach useful, i.e., will learning these clauses ever prevent the solver from exploring an assignment that it wouldn't have otherwise? If so, provide an example; if not, explain why.

10 **Task 2** Given a partial interpretation a clause can be either satisfied, conflicting, unit or unresolved. For the partial interpretation $I = \{a, \neg c, d\}$ identify the status of each of the following clauses:

$(a \vee \neg a)$ $\equiv$ _____

$(\neg a \vee b \vee c)$ $\equiv$ _____

$(b \vee \neg b \vee \neg a)$ $\equiv$ _____

$(\neg d \vee a)$ $\equiv$ _____

$(\neg a \vee b \vee e)$ $\equiv$ _____

4 SAT Encodings (20 points)

20 **Task 1** An isomorphism of undirected graphs G_1 and G_2 is a bijection f between their vertices such that any two vertices $v, v' \in G_1$ are connected by a single edge if and only if $f(v)$ and $f(v')$ are connected by a single edge in G_2. Describe how to encode this as a propositional formula that is satisfiable if and only if G_1 and G_2 are isomorphic.

- Explain how many variables are required, and how the variables are interpreted.
- Likewise, explain which clauses are necessary and what they mean.

Demonstrate your encoding on the graphs shown below. Note that they are not isomorphic, because the second graph does not have a self-edge on the left node.

5 Arrays and Uninterpreted Functions (25 points)

10 **Task 1** Provide a formula in the theory of equality and uninterpreted functions that is valid if an only if the following formula in the theory of arrays is valid:

$$\mathsf{read}\,(\mathsf{write}\,a\,i\,(\mathsf{read}\,b\,j))\,j = x \wedge \mathsf{read}\,b\,i \neq x \wedge i = j$$

If you introduce any uninterpreted functions in your solution, explain what they correspond to in the original formula.

15 **Task 2** Compute the congruence closure of your solution for Task 1, and state whether the congruence classes satisfy the equality and uninterpreted functions formula.

6 Certificates (30 points)

Consider the formula:

$$\underbrace{(\neg p_1 \vee \neg p_2)}_{C_1} \wedge \underbrace{(\neg p_2 \vee p_3)}_{C_2} \wedge \underbrace{(p_1 \vee \neg p_3 \vee \neg p_5)}_{C_3} \wedge \underbrace{(\neg p_5 \vee p_2)}_{C_4} \wedge \underbrace{(p_5 \vee p_2)}_{C_5} \wedge \underbrace{(p_1 \vee \neg p_3 \vee p_5)}_{C_6}$$

10 **Task 1** Which of the following are correct clausal certificates for this formula? Explain your answer in terms of the reverse unit propagation property.

(5 points) $[p_5 \vee p_2, \neg p_5 \vee p_2, \bot]$

(5 points) $[\neg p_1, \neg p_2, \bot]$

10 **Task 2** Recall that resolution certificates are composed of a list of proof steps, which are of the form:

$$\begin{array}{lll} \mathit{Step}\ \#: & \textbf{Assume} & C \\ \mathit{Step}\ \#: & \textbf{Resolve} & C \quad [\mathit{Step}\ \#, \mathit{Step}\ \#, \ldots] \end{array}$$

The **Resolve** steps give the result C of applying resolution on the sequence of clauses, identified by step numbers, obtained at earlier steps. The last step should be $\bot$.

Explain how to obtain a resolution certificate from a clausal certificate. That is, explain how each step of the clausal proof corresponds to a sequence of resolution steps involving clauses from the original formula, as well as earlier clauses in the certificate.

10 **Task 3** Provide a resolution certificate corresponding to the clausal certificate $[p, \bot]$ on the following formula:

$$\underbrace{(p \vee q)}_{C_1} \wedge \underbrace{(\neg p \vee q)}_{C_2} \wedge \underbrace{(\neg r \vee \neg q)}_{C_3} \wedge \underbrace{(r \vee \neg q)}_{C_4}$$

7 Temporal Logic (30 points)

15 **Task 1** Draw a Kripke structure that satisfies the formula $\mathbf{A}[a\,\mathbf{U}\,\mathbf{AF}\,b] \wedge \mathbf{EX}\,\neg b$.

15 **Task 2** For each state in your answer to Task 1, label which of the formulas $\mathbf{AF}\,b$, $\mathbf{EX}\,\neg b$, and $\mathbf{A}[a\,\mathbf{U}\,\mathbf{AF}\,b]$ are satisfied. You may refer to them as P, Q, and R, respectively.