CS 245 Final Exam Practice Questions - Answers

Peter He

Fall 2019

1 Structural Induction

Question 2c) incomplete.

1. a) Let X be the set of all triplets of natural numbers (a, b, c). Let $A = \{(13, 15, 26)\}$. Let P be the set of the operations

$$\{(a,b,c) \mapsto (a-1, b-1, c+2), (a,b,c) \mapsto (a-1, b+2, c-1), (a,b,c) \mapsto (a+2, b-1, c-1)\}$$

Let the set PebblePiles be $I(X, A, P)$.

b)

Proof. Basis: Consider $15 - 13 = 2$. Trivially, $2 \equiv 2 \pmod 3$.
Induction Hypothesis: Let (a, b, c) be a triplet such that $(b - a) \equiv 2 \pmod 3$.
Induction Step: We go through each of the operations in P:

- Consider the triplet $(a - 1, b - 1, c + 2)$. We have that

$$\begin{aligned}(b-1)-(a-1) &= b-a-1+1 \\ &= b-a \\ &\equiv 2 \pmod 3 \qquad \text{by IH}\end{aligned}$$

- Consider the triplet $(a - 1, b + 2, c - 1)$. We have that

$$\begin{aligned}(b+2)-(a-1) &= b-a+2+1 \\ &= b-a+3 \\ &\equiv 2 \pmod 3 \qquad \text{by IH}\end{aligned}$$

- Consider the triplet $(a + 2, b - 1, c - 1)$. Similarly, $(b - 1) - (a + 2) \equiv 2 \pmod 3$ by IH.

By the principle of structural induction, $(b - a) \equiv 2 \pmod 3$ for every element $(a, b, c) \in$ PebblePiles. ∎

c) We would like to prove that there does not exist an element $x \in$ PebblePiles such that x is of the form:

- $(n, 0, 0)$,
- $(0, n, 0)$, or
- $(0, 0, n)$ for some $n \in \mathbb{Z}_{\geq 1}$.

Proof. Let (a, b, c) be an element in PebblePiles. It can be shown by structural induction that $(c - a) \equiv 1 \pmod 3$ and $(c - b) \equiv 2 \pmod 3$. So $a \neq b, b \neq c, c \neq a$. Thus, (a, b, c) cannot be of the above forms. ∎

2. a) Let $\mathbb{A} = \{p_i : i \in \mathbb{Z}_{\geq 1}\}$ and

$$P \left\{ \frac{x, y}{x \vee x}, \frac{x, y}{x \wedge y}, \frac{x, y}{x \to y} \right\}$$

.

b)

Proof. Basis: Let $A = p_i$ for some $i \in \mathbb{Z}_{\geq 1}$. By construction, $A^{v_1} = (p_i)^{v_1} = 1$.
Induction Hypothesis: Let $A, B \in P_{NoNot}$, and assume $A^{v_1} = B^{v_1} = 1$.
Induction Step: We go through each element in P.

- Consider $A \vee B$. By the IH and the truth table of $\vee$, $(A \vee B)^{v_1} = 1$
- Consider $A \wedge B$. By the IH and the truth table of $\wedge$, $(A \wedge B)^{v_1} = 1$
- Consider $A \to B$. By the IH and the truth table of $\to$, $(A \to B)^{v_1} = 1$

■

c)

Proof. First, we construct a truth table for $A = (p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2)$.

p_1	p_2	$(p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2)$
1	1	0
1	0	1
0	1	1
0	0	0

By this truth table, $A = \neg(p_1 \leftrightarrow q)$.
Let $I(X, \mathbb{A}', P) \subset P_{NoNot}$ where, WLOG,

$$\mathbb{A}' \subseteq \mathbb{A}, \mathbb{A}' = \{p_1 \wedge p_2, p_2 \wedge p_1, p_1 \vee p_2, p_2 \vee p_1, p_1 \to p_2, p_2 \to p_1\}$$

. We go by structural induction on this set.
Basis: None of the truth tables for $\to$, $\wedge$, and $\vee$ are the same as A.
Induction Hypothesis: Assume $\alpha, \beta \in I(X, \mathbb{A}', P)$ are formulas that are not tautologically equivalent to A.
Induction Step: We go through each element in P.

- Consider $\alpha \wedge \beta$. For the sake of contradiction assume $\alpha \wedge \beta$ is tautologically equivalent to A. This implies, that for a valuation t such that:
  - $p_1^t = p_2^t = 1$, $\alpha^t = 0$ or $\beta^t = 0$, and
  - p_1

■

2 Formal Proofs in Propositional Logic

1. Basis ($n = 1$): We wish to show $\{(A_1 \to A_2)\} \vdash (A_1 \to A_2)$, which is a one line proof by ($\in$).
   Inductive Hypothesis: Assume that $\{(A_1 \to A_2), \ldots, (A_{n-1} \to A_n)\} \vdash (A_1 \to A_n)$.
   Induction Step: We wish to show $\{(A_1 \to A_2), \ldots, (A_n \to A_{n+1})\} \vdash (A_1 \to A_{n+1})$.

   Proof.

$$\{(A_1 \to A_2), \ldots, (A_n \to A_{n+1})\} \vdash (A_1 \to A_n) \quad \text{by IH} \quad (1)$$
$$\{A_1, (A_1 \to A_2), \ldots, (A_n \to A_{n+1})\} \vdash (A_1 \to A_n) \quad \text{by } (+, 1) \quad (2)$$
$$\{A_1, (A_1 \to A_2), \ldots, (A_n \to A_{n+1})\} \vdash A_1 \quad \text{by } (\in) \quad (3)$$
$$\{A_1, (A_1 \to A_2), \ldots, (A_n \to A_{n+1})\} \vdash A_n \quad \text{by } (\to -, 2, 3) \quad (4)$$
$$\{A_1, (A_1 \to A_2), \ldots, (A_n \to A_{n+1})\} \vdash (A_n \to A_{n+1}) \quad \text{by } (\in) \quad (5)$$
$$\{A_1, (A_1 \to A_2), \ldots, (A_n \to A_{n+1})\} \vdash A_{n+1} \quad \text{by } (\to -, 4, 5) \quad (6)$$
$$\{(A_1 \to A_2), \ldots, (A_n \to A_{n+1})\} \vdash (A_1 \to A_{n+1}) \quad \text{by } (\to +, 6) \quad (7)$$

■

2. $\underline{\rightarrow}$: Assume $\vdash (A_1 \rightarrow (A_2 \rightarrow (A_3 \rightarrow A_4)))$. Using Gao's/Collin's system, we have:

$$
\begin{aligned}
&\vdash (A_1 \rightarrow (A_2 \rightarrow (A_3 \rightarrow A_4))) && && (8)\\
A_1 &\vdash (A_1 \rightarrow (A_2 \rightarrow (A_3 \rightarrow A_4))) && \text{by } (+, 8) && (9)\\
A_1 &\vdash (A_2 \rightarrow (A_3 \rightarrow A_4)) && \text{by } (\rightarrow -, 9) && (10)\\
A_1, A_2 &\vdash (A_2 \rightarrow (A_3 \rightarrow A_4)) && \text{by } (+, 10) && (11)\\
A_1, A_2 &\vdash (A_3 \rightarrow A_4) && \text{by } (\rightarrow -, 11) && (12)\\
A_1, A_2, A_3 &\vdash (A_3 \rightarrow A_4) && \text{by } (+, 12) && (13)\\
A_1, A_2, A_3 &\vdash A_4 && \text{by } (\rightarrow -, 13) && (14)\\
A_1, A_2, A_3 &\vdash A_2 && \text{by } (\in) && (15)\\
A_1, A_3 &\vdash (A_2 \rightarrow A_4) && \text{by } (\rightarrow +, 14, 15) && (16)\\
A_1, A_3 &\vdash A_1 && \text{by } (\in) && (17)\\
A_3 &\vdash (A_1 \rightarrow (A_2 \rightarrow A_4)) && \text{by } (\rightarrow +, 16, 17) && (18)\\
A_3 &\vdash A_3 && \text{by } (\in) && (19)\\
&\vdash (A_3 \rightarrow (A_1 \rightarrow (A_2 \rightarrow A_4))) && \text{by } (\rightarrow +, 18, 19) && (20)
\end{aligned}
$$

$\underline{\leftarrow}$: Assume $\vdash (A_3 \rightarrow (A_1 \rightarrow (A_2 \rightarrow A_4)))$. Using Shai's deduction theorem, we have

- $\vdash (A_3 \rightarrow (A_1 \rightarrow (A_2 \rightarrow A_4)))$ if and only if
- $A_3 \vdash\rightarrow (A_1 \rightarrow (A_2 \rightarrow A_4))$ if and only if
- $A_1, A_3 \vdash (A_2 \rightarrow A_4)$ if and only if
- $A_1, A_2, A_3 \vdash A_4$ if and only if
- $A_1, A_2 \vdash (A_3 \rightarrow A_4)$ if and only if
- $A_1 \vdash (A_2 \rightarrow (A_3 \rightarrow A_4))$ if and only if
- $\vdash (A_1 \rightarrow (A_2 \rightarrow (A_3 \rightarrow A_4)))$

3. Let $\beta = (\neg a) \wedge b \wedge c$. $\Sigma \not\models \beta$ since there is a valuation t such that $\Sigma^t = 1$ and $\beta^t = 0$. Specifically, define t such that

$$a^t = b^t = c^t = 1$$

. By soundness of propositional logic, $\Sigma \nvdash \beta$.

4. *Proof.* Let t be a valuation such that $\Sigma^t = 1$. For the sake of contradicton, assume $c^t = 0$. Then $(\neg a)^t = 0$ since $(\neg a \rightarrow c)^t = 1$, which implies $a^t = 1$. This further implies that $b^t = 1$ since $(a \rightarrow b)^t = 1$. However, $b^t = 0$ since $(b \rightarrow c)^t = 1$. So $c^t = 1$. By the truth table of $\rightarrow$, $(\neg c \rightarrow \gamma)^t = 1$ for any formula γ. Thus, $\Sigma \models \gamma$, and by completeness of propositional logic, $\Sigma \vdash \gamma$. ■

3 Consistency and Satisfiability of sets of formulas

1. *Proof.* Assume Σ is satisfiable. Then there is a valuation t such that $\Sigma^t = 1$. There are two cases:

   - Assume $\Sigma \cup \{\alpha\}$ is not satisfiable. Then $\alpha^t = 0$ and $(\neg\alpha)^t = 1$. Thus, $\Sigma \cup \{\neg\alpha\}$ is satisfiable. Specifically, it is satisfied by t.
   - Similarly, if $\Sigma \cup \{\neg\alpha\}$ is not satisfiable, $\Sigma \cup \{\alpha\}$ is satisfiable.

   ■

2. Note that all propositional formulas are finitely long. Since α is over a finite amount of atoms, let $\Sigma = \{u \rightarrow \alpha\}$, $u = p_i$ not occuring in α.

   - If α is satisfiable or a tautology, then choose a valuation t such that $u^t = 1, \alpha^t = 0$. $\Sigma \cup \{\alpha\}$ is then not satisfiable, and by 1, $\Sigma \cup \{\neg\alpha\}$ is satisfiable.
   - If α is a contradiction, choose a valuation t such that t such that $u^t = 1, \alpha^t = 1$. $\Sigma \cup \{\neg\alpha\}$ is then not satisfiable, by 1, $\Sigma \cup \{\alpha\}$ is satisfiable.

3. Use the same set from 2.

4. *Proof.* Tutorial 5 Question 1. ■

5. *Proof.* Tutorial 5 Question 2. ■

6. It is not always the case. Consider $\Sigma = \{p\}, \Sigma' = \{\neg p\}$, and $\beta = p \wedge p$. $\Sigma \cup \Sigma'$ is clearly inconsistent. $\Sigma \vdash \beta$ by $\wedge+$. $\Sigma' \models \neg\beta$, which can be shown by a truth table, so $\Sigma' \vdash \neg\beta$ by completeness of propositional logic. However, $\beta \notin \Sigma \cup \Sigma'$.

7. (Assignment 4 Question 2a) Let $\Sigma' = \{p, \neg q, (\neg p \vee q)\}$. For each pair $(A, B) \in \Sigma'$, it can be shown that $\{A, B\}$ is satisfiable.

   - $\{p, \neg q\}$, choose t such that $p^t = 1, q^t = 0$.
   - $\{p, (\neg p \vee q)\}$, choose t such that $p^t = 1 = q^t = 1$.
   - $\{\neg q, (\neg p \vee q)\}$, choose t such that $p^t = q^t = 0$.

   By soundness of propositional logic, each of these sets are consistent.
   However, it can be shown by truth table that Σ' is not satisfiable. By completeness of propositional logic, Σ' is not consistent.

8. For $k = 3$, we can let $\Sigma'' = \{p, q, \neg r, (\neg p \vee \neg q \vee r)\}$. It is clear that each pair can form a satisfiable set. It can also be shown by truth table that Σ'' is not satisfible.
   For $k \geq 2$, let $\Sigma'' = \{p_1, \ldots, p_{k-1}, \neg p_k, (\neg p_1, \ldots, \neg p_{k-1}, p_k)\}$.

9. The last claim does not contradict the statement since any Σ'' we create is finite, so a finite inconsistent subset we can find is Σ'' itself.

10. (a) (Assignment 4 Question 2b) The statement is true.

    Proof. Assume Σ is satisfiable and $\Sigma \models \alpha$. By definition, for any truth valuation t such that $\Sigma^t = 1$, we have $\alpha^t = 1$. Since Σ is satisfiable, such a valuation exists. So we can choose any valuation that satisfies Σ, and it will satisfy $\Sigma \cup \{\alpha\}$. ■

    (b) The statement is true.

    Proof. Assume Σ is consistent and $\Sigma \vdash \alpha$. By completeness and soundness of propositional logic, Σ is satisfiable and $\Sigma \models \alpha$. By a), $\Sigma \cup \{\alpha\}$ is satisfiable. By soundness again, $\Sigma \cup \{\alpha\}$ is consistent. ■

4 Decidability

Question 3 and 4 incomplete.

1. W_1 is not decidable. I assume that the question means σ is code for a program that could possibly take in many inputs, but will halt on at least one input.

   Proof. For the sake of contradiction, assume there is an algorithm `B` which solves this problem. We will construct an algorithm `A` which solves the halting problem. Algorithm `A` works as follows.

   - `A` takes in two inputs, a program `P` and an input `I`.
   - Let program `P'` run the program `P`, and return `P()`.
   - Run algorithm `B` with the code of `P'`, σ, as input.

   ■

2. W_2 is decidable.

   Proof. Note that σ is finitely long, and that σ is valid C code that compiles. We can construct an algorithm `B` for input σ that makes W_1 decidable. `B` works as follows:

   - Convert binary code σ to regular C code.

- Use the regex `if.*(.*).*{.*}.*(else|else.*if).*{.*}` to match any strings in the code.
- If the regex matches with any string, halt and return 1.
- Otherwise, return 0.

The above will always halt since σ is finite. ■

3. For clarification, σ_1 and σ_2 are inputs for the program σ. W_3 is not decidable.

   Proof. For the sake of contradiction, assume there is a program `B` that solves this problem. We will construct an algorithm `A` that solves the halting problem, which works as follows:

   - `A` takes in two inputs, a program `P` and an input `I`

   ■

4. wuhh

5. W_5 is decidable. An algorithm can take in σ as input, count the number of bits $s = \#(\sigma)$, and return $s \equiv 0 \pmod 2$.

6. W_6 is decidable. An algorithm can convert σ to base 10 (optional), and run a prime checking program on it. Since σ is finite, this program will always halt.