Assignment 0 Draft Solution

Advanced Computing Concepts for Engineering

Due January 27, 2015

Note that the work that you turn in for this assignment must represent your individual effort. You are welcome to help your fellow students to understand the material of the course and the meaning of the assignment questions, however, the answer that you submit must be created by you alone.

Please consider preparing your assignment with a typesetting program such as TeX, LaTeX, LyX, Scientific Word, or MS Word.

Q0 [8] Propositional logic.

Suppose that, instead of defining conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$) from implication ($\Rightarrow$), we had instead defined implication from or and not with the following definition

$$(p \Rightarrow q) = (\neg p \vee q) \text{ Material implication.}$$

Using the laws in the notes about conjunction, disjunction, and negation, write equational proofs of the following laws

(a) The contrapositive law: $(p \Rightarrow q) = (\neg q \Rightarrow \neg p)$.

Solution:

$$
\begin{array}{ll}
 & p \Rightarrow q \\
= & \qquad \text{“Material implication”} \\
 & \neg p \vee q \\
= & \qquad \text{“Double negation”} \\
 & \neg p \vee \neg\neg q \\
= & \qquad \text{“Commutativity”} \\
 & \neg\neg q \vee \neg p \\
= & \qquad \text{“Material implication”} \\
 & \neg q \Rightarrow \neg p
\end{array}
$$

(b) Shunting: $(p \wedge q \Rightarrow r) = (p \Rightarrow (q \Rightarrow r))$

Solution:

$$
\begin{array}{ll}
 & p \wedge q \Rightarrow r \\
= & \qquad \text{“Material implication”} \\
 & \neg (p \wedge q) \vee r \\
= & \qquad \text{“De Morgan's law”} \\
 & (\neg p \vee \neg q) \vee r \\
= & \qquad \text{“Associativity”} \\
 & \neg p \vee (\neg q \vee r) \\
= & \qquad \text{“Material implication”} \\
 & \neg p \vee (q \Rightarrow r) \\
= & \qquad \text{“Material implication”} \\
 & p \Rightarrow (q \Rightarrow r)
\end{array}
$$

Q1 [10] Substitutions

(a) Underline all bound occurrences of variables in the following formulae. Circle all free occurrences of variables.

$$\{i \in \mathbb{N} \mid \text{prime}\,(i) \wedge \text{prime}(i+2) \cdot (i, i+2)\}$$

$$(\forall i \in \{j, ..j+10\} \cdot f(j) = f(i))$$

Solution:

$$\{\underline{i} \in \mathbb{N} \mid \text{prime}\,(\underline{i}) \land \text{prime}(\underline{i} + 2) \cdot (\underline{i}, \underline{i} + 2)\}$$

$$\left(\forall \underline{i} \in \{\boxed{j}, ..\boxed{j} + 10\} \cdot \boxed{f}(\boxed{j}) = \boxed{f}(\underline{i})\right)$$

(b) Make the following substitutions

$$\{i \in \mathbb{N} \mid \text{prime}\,(i) \land \text{prime}(i + 2) \cdot (i, i + 2)\}\,[i : i + 1]$$

$$(\sum_{i \in \{j,..k\}} f(i) = \sum_{i \in \{j,..k\}} g(i))[j : j + 1]$$

$$(\sum_{i \in \{j,..k\}} f(i) = \sum_{i \in \{j,..k\}} g(i))[j : i]$$

Solution:

$$\{i \in \mathbb{N} \mid \text{prime}\,(i) \land \text{prime}(i + 2) \cdot (i, i + 2)\}$$

$$(\sum_{i \in \{j+1,..k\}} f(i) = \sum_{i \in \{j+1,..k\}} g(i))$$

$$(\sum_{m \in \{i,..k\}} f(m) = \sum_{m \in \{i,..k\}} g(m))$$

Q2 [12] Quantifiers and sets

Let P be the set of all people on a social network and let friend : $P \times P \rightarrow \mathbb{B}$ be a boolean function expressing that the first person is a friend of the second.

(a) Use quantifiers to say that friendship is symmetric, i.e. everyone is a friend of all their friends.

Solution: $\forall x, y \in P \cdot \text{friend}(x, y) = \text{friend}(y, x)$ or $\forall x, y \in P \cdot \text{friend}(x, y) \Rightarrow \text{friend}(y, x)$

(b) A *clique* is a set of people who are all friends with each other. Use quantifiers to express that a set $S \subseteq P$ is a clique.

Solution: $\forall x, y \in S \cdot \text{friend}(x, y)$

(c) Explain the meaning of the following expression in clear English

$$\forall x \in R \cdot \exists y \in R \cdot \text{friend}(x, y)$$

Solution: Everyone in R is a friend of someone in R.

(d) Explain the meaning of the following expression in clear English

$$\exists x \in R \cdot \forall y \in R \cdot \text{friend}(x, y)$$

Solution: There is someone in R who is a friend of everyone in R (including themselves).

Q3 [10] Refinement

(a)[5] Make a table of all (9) behaviours belonging to $\Sigma \dagger \Sigma$ where

$$\Sigma = \{\text{“}x\text{”} \mapsto \{1, 2, 3\}\}$$

For each behaviour, indicate whether it is accepted (✓) or rejected (×) by each of the following specifications (on $\Sigma \dagger \Sigma$)

$$\begin{aligned}
a &= \langle x' = 2 \rangle \\
b &= \langle x \geq 2 \Rightarrow x' = 2 \rangle \\
c &= \langle x' = 1 \vee x' = 2 \rangle \\
d &= \langle x \geq 2 \Rightarrow (x' = 1 \vee x' = 2) \rangle \\
\textbf{magic} &= \langle \mathfrak{false} \rangle \\
\textbf{abort} &= \langle \mathfrak{true} \rangle
\end{aligned}$$

Solution: Checkmarks indicate accepted behaviours

x	x'	**magic**	a	b	c	d	**abort**
1	1			✓	✓	✓	✓
1	2		✓	✓	✓	✓	✓
1	3			✓		✓	✓
2	1				✓	✓	✓
2	2		✓	✓	✓	✓	✓
2	3						✓
3	1				✓	✓	✓
3	2		✓	✓	✓	✓	✓
3	3						✓

(b) [5] Look up Hasse diagrams in the Wikipedia. Draw a Hasse diagram illustrating all the refinement relationships between these specifications.

Solution:

$$\begin{tikzcd}[arrows=dash]
 & \textbf{abort} \arrow[d] & \\
 & d \arrow[dl] \arrow[dr] & \\
b \arrow[dr] & & c \arrow[dl] \\
 & a \arrow[d] & \\
 & \textbf{magic} &
\end{tikzcd}$$

Q4 [4] Implementability Again $\Sigma = \{\text{“}x\text{”} \mapsto \{0,1,2\}\}$
Consider the following specifications on $\Sigma \dagger \Sigma$

$$\begin{aligned}
f &= \langle x = 0 \Rightarrow x' > 0\rangle \\
g &= \langle x = 0 \wedge x' > 0\rangle \\
h &= \langle x' = (x+1) \bmod 3\rangle \\
\textbf{magic} &= \langle \mathfrak{false}\rangle \\
\textbf{abort} &= \langle \mathfrak{true}\rangle
\end{aligned}$$

Which of these specifications are implementable? Explain why.

Solution:

- f is implementable since whatever the input, an output of $\{\text{“}x\text{”} \longmapsto 1\}$ or $\{\text{“}x\text{”} \longmapsto 2\}$.
- g is unimplementable since when the input is not 0 there is no output that makes an acceptable behaviour.
- h is implementable since for each input there is one output.
- **magic** is unimplementable since there is never an acceptable output.
- **abort** is implementable since for every input every output is acceptable.

Q5 [10] Specification

Suppose $n > 0$ is a fixed member of $\mathbb{N}$. Remember that $\{0, ..n\}$ is the set containing the first n natural numbers. Then $\{0, ..n\} \xrightarrow{\text{tot}} \mathbb{R}$ is the set of all *sequences* of real numbers with a *length* of n. As an abbreviation we'll write $\mathbb{R}^n$ for $\{0, ..n\} \xrightarrow{\text{tot}} \mathbb{R}$. We can use $\mathbb{R}^n$ as the type of real-valued arrays of length n. Of course, if $a \in \mathbb{R}^n$ and $i \in \{0, ..n\}$, then $a(i)$ is item i of array a. The expression $a(i)$ is not defined if $a \in \mathbb{R}^n$ but $i \notin \{0, ..n\}$.

Let

$$\Sigma = \{\text{“}a\text{”} \mapsto \mathbb{R}^n, \text{“}x\text{”} \mapsto \mathbb{R}, \text{“}i\text{”} \mapsto \mathbb{N}\}$$

Write specifications on $\Sigma \dagger \Sigma$ for the problems below. The following function will be helpful

$$\begin{array}{rcl} \text{count} & \in & \mathbb{R}^n \times \mathbb{R} \xrightarrow{\text{tot}} \mathbb{N} \\ \text{count}(a, x) & = & |\{j \in \{0, ..n\} \mid a(j) = x\}| \end{array}$$

(a) [5] Search: The final value of i indicates a location in a where x can be found, if there is one. If x is not at any location in a, then the final value of i should be n. In either case neither a nor x change.

$a = a' = [7, 4, 7, 0, 4]$	$a = a' = [7, 4, 7, 0, 4]$	$a = a' = [7, 4, 7, 0, 4]$
$x = x' = 4$	$x = x' = 4$	$x = x' = 2$
$i =$ anything	$i =$ anything	$i =$ anything
$i' = 4$	$i' = 1$	$i' = 5$

Solution:

$$\left\langle \begin{array}{ll} & (\text{count}(a, x) = 0 \Rightarrow i' = n) \\ \wedge & (\text{count}(a, x) > 0 \Rightarrow a(i') = x) \\ \wedge & a' = a \wedge x' = x \end{array} \right\rangle$$

or

$$\left\langle \begin{array}{ll} & \left(\begin{array}{ll} & (\text{count}(a, x) = 0 \wedge i' = n) \\ \vee & (\text{count}(a, x) > 0 \wedge a(i') = x) \end{array} \right) \\ \wedge & a' = a \wedge x' = x \end{array} \right\rangle$$

(b) [5] Least: The final value of x is the smallest value in a, and the final value of i indicates (one of) its location(s). a does not change. (You may assume $n > 0$.) For example, any of the following two behaviours is accepted.

$$\begin{array}{ll} a = a' = [7, 4, 7, 0, 4] & \\ x = \text{anything} & x' = 0 \\ i = \text{anything} & i' = 3 \end{array}$$

Solution:

$$\left\langle \begin{array}{ll} & x' = a(i') \\ \wedge & a' = a \\ \wedge & \forall j \in \{0, ..n\} \cdot a(j) \geq x' \end{array} \right\rangle$$