198:205
Discrete Structures I
Professor McCarty

Solutions to Sample Final Exam

Question 1:

The set identity in part (a) is false. For a counter example, let $A = \{1, 2\}$, $B = \{1\}$ and $C = \{2\}$. Then:

$$A - (B \cap C) \;=\; A - (\{1\} \cap \{2\}) \;=\; A - \emptyset \;=\; A,$$

but

$$(A - B) \cap (A - C) \;=\; \{2\} \cap \{1\} = \emptyset.$$

The set identity in part (b) is true. Here is a proof using a chain of logical equivalences:

$$\begin{aligned} A - (B \cap C) &= \{x \mid x \in A \;\wedge\; \neg(x \in B \cap C)\}, \text{ by definition of set difference,} \\ &= \{x \mid x \in A \;\wedge\; \neg(x \in B \wedge x \in C)\}, \text{ by definition of } \cap, \\ &= \{x \mid x \in A \;\wedge\; (\neg x \in B \vee \neg x \in C)\}, \text{ by De Morgan's law,} \\ &= \{x \mid (x \in A \wedge \neg x \in B) \;\vee\; (x \in A \wedge \neg x \in C)\}, \text{ by distributive law,} \\ &= \{x \mid x \in (A - B) \;\vee\; x \in (A - C)\}, \text{ by definition of set difference,} \\ &= (A - B) \cup (A - C), \text{ by definition of } \cup. \end{aligned}$$

Here is a proof using a chain of set identities:

$$\begin{aligned} A - (B \cap C) &= A \cap (\overline{B \cap C}), \text{ by definition of set difference,} \\ &= A \cap (\overline{B} \cup \overline{C}), \text{ by De Morgan's law for sets,} \\ &= (A \cap \overline{B}) \cup (A \cap \overline{C}), \text{ by distributive law for sets,} \\ &= (A - B) \cup (A - C), \text{ by definition of set difference.} \end{aligned}$$

Question 2:

For part (a), assume f is not one-to-one. Then, by definition, there is some $x \in A$ and some $y \in A$, with $x \neq y$, such that $f(x) = f(y)$. Define $A_1 = \{x\}$ and $A_2 = \{x, y\}$. Then $A_1 \subset A_2$ but $f(A_1) = f(A_2)$, which shows that f does not preserve proper subsets.

We have thus shown the contrapositive of the proposition that we were asked to prove. QED.

For part (b), assume f does not preserve proper subsets. Then, by definition, there is some $A_1 \subset A_2 \subseteq A$ for which $f(A_1) = f(A_2)$. Choose some $y \in (A_2 - A_1)$. Then $f(y) \in f(A_2) = f(A_1)$, so there must be some $x \in A_1$ such that $f(x) = f(y)$. But if $x \in A_1$ and $y \in (A_2 - A_1)$, then $x \neq y$, which shows that f is not one-to-one. We have thus shown the contrapositive of the proposition that we were asked to prove. QED.

Question 3:

For the basis step, the expression is $2^0 - 1 = 1 - 1 = 0$, which is divisible by 3. For the inductive step, we assume, for an arbitrary $k \geq 0$, that $2^{2k} - 1$ is divisible by 3, which means that $2^{2k} - 1 = 3m$ for some integer m, or $2^{2k} = 3m + 1$. Now consider the same expression for $k + 1$:

$$\begin{aligned} 2^{2(k+1)} - 1 &= 2^{2k+2} - 1 \\ &= 2^2 2^{2k} - 1 \\ &= 2^2(3m + 1) - 1, \quad \text{by the inductive hypothesis,} \\ &= 12m + 4 - 1 \\ &= 3(4m + 1), \end{aligned}$$

which is obviously divisible by 3. QED.

Question 4:

Note that the proposed loop invariant, P, is a biconditional. For part (a), since `i` is initialized to `1` and `j` is initialized to `n`, P is an identity when execution reaches the `while` loop, and thus trivially true.

For part (b), we assume that P and C are true prior to the execution of the `begin`/`end` block, and we assume that the `begin`/`end` block terminates normally. This means that the `return` statement is not executed, since that would be an abnormal termination. There are thus two cases to consider, corresponding to the two remaining branches of the `if/then/else` statement.

Suppose `A[i] + B[j] < X`. If there are two indices in the ranges [`1`, `m`] and [`1`, `n`] that solve the problem and yield a sum equal to `X`, then, by the truth of P prior to the `begin`/`end` block, there are two indices in the ranges [`i`, `m`] and [`1`, `j`] that also solve the problem. But `B[j]` is the maximum value of `B` in this range, while `A[i]` is the minimum value of `A`, since the arrays are nondecreasing. Since `A[i] + B[j] < X`, we need to increase the value of either `A[i]` or `B[j]`, or both, to achieve the result `A[i] + B[j] =`

`X`. But there is no v in the range $\mathtt{1} \leq v \leq \mathtt{j}$ that can make $\mathtt{B}[v]$ greater than `B[j]`, and if there is a u in the range $\mathtt{i} \leq u \leq \mathtt{m}$ that can make $\mathtt{A}[u] + \mathtt{B}[v] = \mathtt{X}$, it must be greater than `i`. Therefore, when we increment `i` to `i+1` inside the first `else` clause, we cannot change the truth value of the second proposition in P, assuming that it was initially true. Conversely, if the truth value of the second proposition in P was false when we entered the `begin`/`end` block, then it obviously remains false when we increment `i` to `i+1` and restrict the range of the variable u even further. We have thus shown that the proposed loop invariant, P, remains true in the case `A[i] + B[j] < X`.

The case `A[i] + B[j] > X` is similar. In this case, since `A[i]` is the minimum value of `A` in the range $[\mathtt{i}, \mathtt{m}]$ and `B[j]` is the maximum value of `B` in the range $[\mathtt{1}, \mathtt{j}]$, if the second proposition in P was true before execution entered the the `begin`/`end` block, then it remains true when we decrement `j` to `j-1`. Thus the proposed loop invariant, P, remains true in this case as well.

For part (c), we use the inference rule for `while` statements:

$$\frac{(P \wedge C)\ \{S\}\ P}{P\ \{\mathtt{while}\ C\ S\}\ (\neg C \wedge P)}$$

Part (b) established the premise in this rule: $(P \wedge C)\ \{S\}\ P$. Part (a) established the initial assertion in the conclusion: $P\ \{\mathtt{while}\ C\ S\}\ (\neg C \wedge P)$. The final assertion in the conclusion is used in the following argument:

(i) Assume that the `while` loop terminates normally, and the procedure returns `(0,0)`. Then $\neg C \wedge P$ is true, where $\neg C$ is:

$$\mathtt{i} > \mathtt{m} \text{ or } \mathtt{j} < \mathtt{1}.$$

But if `i > m`, then the second proposition in P is false, since

$$\neg \exists u\ [\ \mathtt{i} \leq u \leq \mathtt{m}\ ],$$

and if `j < 1`, then the second proposition in P is false, since

$$\neg \exists v\ [\ \mathtt{1} \leq v \leq \mathtt{j}\ ].$$

Since the biconditional, P, is true, however, the first proposition in P must be false as well, i.e., there are no indices u and v such that $\mathtt{A}[u] + \mathtt{B}[v] = \mathtt{X}$. QED.

(ii) Conversely, assume that there are no indices `i` and `j` such that `A[i]` $+$ `B[j]` $=$ `X`. Then the abnormal exit, `return (i,j)`, cannot be executed. But the procedure still terminates. To see this, note that each time around the `while` loop either `i` is incremented or `j` is decremented, so eventually either the condition `i <= m` or the condition `j >= 1` is violated. Thus the `while` loop terminates normally, and the procedure returns `(0,0)`. QED.

Question 5:

Assume that R is reflexive and euclidean. To show that R is symmetric, choose an arbitrary $(a, b) \in R$. By reflexivity, $(a, a) \in R$, and thus by the euclidean property, $(b, a) \in R$. QED.

To show that R is transitive, assume that $(a, b) \in R$ and $(b, c) \in R$, for arbitrary $a, b, c \in A$. By the symmetry property just proven, $(b, a) \in R$. Therefore, by the euclidean property, $(a, c) \in R$. QED.

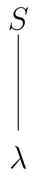
Question 6:

To show that $\mathbf{B} \subseteq L(\mathbf{G})$, we need to show that every balanced string of parentheses can be generated by the grammar $\mathbf{G}$. We will prove this by strong induction on the length of a string. The basis step is obviously true, since the string of length 0 is balanced and can be generated in one rewriting by the rule $S \rightarrow \lambda$. For the inductive step, we choose an arbitrary $n \geq 0$, and we assume that every balanced string of length $\leq 2n$ can be derived using the grammar $\mathbf{G}$. (Notice that the length of a balanced string is always even.) Now consider a balanced string, w, of length $2(n+1)$. Obviously, w must begin with a left parenthesis, and we can move along the string to the right until we find the "matching" right parenthesis, that is, we can find the smallest initial segment of w that has the same number of left and right parentheses overall. This initial segment will have the form (x), and we can write $w = (x)\,y$. By a check of the definition, we can verify that x and y are both balanced strings. Thus, by the inductive hypothesis, since x and y have length $\leq 2n$, there exist derivations in $\mathbf{G}$ of $S \overset{*}{\Longrightarrow} x$ and $S \overset{*}{\Longrightarrow} y$. But we can now construct a new derivation:

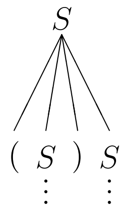
$$\begin{aligned} S &\Longrightarrow (S)\,S \\ &\overset{*}{\Longrightarrow} (x)\,S \\ &\overset{*}{\Longrightarrow} (x)\,y, \end{aligned}$$

which is a derivation of w, as required. QED.

To show that $L(\mathbf{G}) \subseteq \mathbf{B}$, we need to show that every terminal derivation in the grammar $\mathbf{G}$ results in a balanced string of parentheses. We will prove this by strong induction on the height of a derivation tree. For the basis step, the only derivation of a terminal string in a tree of height 1 is:



and the empty string is obviously balanced. For the inductive step, we choose an arbitrary $n \geq 1$, and we assume that every terminal derivation tree of height $\leq n$ yields a balanced string of parentheses. Consider a terminal derivation tree of height $n+1$. It must have the form:



where each of the lower S nodes is the root of a terminal derivation tree of height $\leq n$. By the inductive hypothesis, these derivation trees both yield balanced strings, say x and y, respectively. However, if x and y are balanced, then $(x)\,y$ is balanced, by a simple check of the definition. Thus the terminal derivation tree of height $n+1$ yields a balanced string of parentheses, as required. QED.