CS 320L – Applied Discrete Mathematics – Spring 2012
Instructor: Marc Pomplun

Assignment #1

Posted on February 6 – Due by February 21, 5:30pm

Question 1: Formalization of Logical Expressions

a) Franz and Erika are the only professors who play soccer.

b) All CS320 students know each other and each other's spouses.

c) Petra and Bert never take classes that are interesting.

d) Any UMB student will fail if he/she does not complete Assignment #1.

e) Andreas beats up all professors that fail him and do not fail his sister Susanne.

f) There is at most one computer scientist who can dance.

g) There is one UMB student who is taller than 7 feet and taller than all other UMB students.

h) All UMB students who work full-time prefer evening classes.

i) There are no scientists who are neither crazy nor dangerous.

j) George and Claudia always attend the same courses.

Question 2: Tautologies and Contradictions

Find out for each of the following propositions whether it is a tautology, a contradiction, or neither (a contingency). Prove your answer.

a) $[(p \rightarrow q) \wedge (q \rightarrow p)] \leftrightarrow (p \leftrightarrow q)$

b) $(p \wedge q \wedge r) \rightarrow [(q \rightarrow r) \vee (p \rightarrow q)]$

Question 3: Set Operations

Let us take a look at the sets $A = \{x, y, z\}$, $B = \{1, 2\}$, $C = \{x, z\}$. List the elements of the following sets D, E, F, G, H, and I:

a) $D = (A \times B) - (C \times B)$

b) $E = 2^A - 2^C$

c) $F = 2^{(A - B - C)}$

d) $G = (A \times B \times C) - (C \times B \times A)$

e) $H = \{(a, b, c) \mid a, b, c \in A \wedge a \neq b \wedge a = c\}$.

f) $I = \{(a, b) \mid a \in C \wedge b \in A \wedge b \notin C\}$

Question 4: Cardinality

Are the following statements true for all sets A, B and C? Prove your answers.

a) $|A \cup B \cup C| = |A - B - C| + |B - A - C| + |C - A - B|$

b) $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C|$

Question 5: Functions

Find out whether the following functions from **R** to **R** are injective, surjective, and/or bijective.

a) $f(z) = z^2 - 1$

b) $f(z) = 3z - 7$

c) $f(z) = z/(z^2 + 1)$

d) $f(z) = z^2/(z^2 + 1)$

Question 6: Function Proofs

Are the following statements true or false? In each case, prove your answer.

a) For every bijective function *f* with a finite domain, the domain and the range of *f* have identical cardinalities.

b) There is no strictly decreasing function f: $\mathbf{N} \rightarrow \mathbf{N}$.

Question 7: Big-O Estimates

Give as good a big-O estimate as possible for the following complexity functions:

a) $(n^2 + 12n)(\log n - 30)$

b) $(5n! + 3n^3) + (3^n \cdot 4n^2)$

c) $n^2 \cdot (n^3 + 5 \log n) + n^2 (n^2 + 10)$

Question 8: Algorithms and Their Complexity

a) Write a simple bubble sort procedure in pseudocode that receives a sequence of integers $a_1, \ldots, a_n$ as its input and returns a sequence of integers $b_1, \ldots, b_n$ that contains the same values as the input sequence but in ascending order. For example, input 5, 9, 3 would yield the output 3, 5, 9. The basic idea of bubble sort is to start with the input sequence and repeatedly traverse the sequence while comparing consecutive terms (for example, a_2 and a_3) and switching them if the first term is greater than the second one. This is repeated until the complete sequence is traversed without any switches being made – this means that all elements must be listed in ascending order. There are many variants of bubble sort; any of them that perform the sorting task correctly are OK for this question.

b) Describe the kind of input that causes worst-case time complexity for your algorithm for an input sequence of length n (only count comparisons), and explain why this is the case.

c) Provide an equation for your algorithm that describes the number of required comparisons as a function of input length n in the worst case. It may be a good idea to first use a sum notation.

d) Convert your equation from (c) into a closed-form equation, i.e., one that no longer uses the sum symbol but only operations such as multiplication or addition of individual numbers or variables.

e) Use the big-O-notation to describe the worst-case time complexity of your algorithm.