Practice Midterm Exam

Sample Solutions

Question 1:	out of	points
Question 2:	out of	points
Question 3:	out of	points
Question 4:	out of	points
Question 5:	out of	points

Total Score:

Grade:

Question 1: What do we know?

Let p, q, and r be propositions; p is known to be true, q is known to be false, and r's truth value is unknown at this time. Tell whether each of the following compound propositions is true, is false, or has an unknown truth value at this time by circling the appropriate word.

a) $p \vee r$ — (true) — false — unknown

b) $p \wedge r$ — true — false — (unknown)

c) $p \rightarrow r$ — true — false — (unknown)

d) $q \rightarrow r$ — (true) — false — unknown

e) $r \rightarrow p$ — (true) — false — unknown

f) $r \rightarrow q$ — true — false — (unknown)

g) $(p \wedge r) \leftrightarrow r$ — (true) — false — unknown

h) $(p \vee r) \leftrightarrow r$ — true — false — (unknown)

i) $(q \wedge r) \leftrightarrow r$ — true — false — (unknown)

j) $(q \vee r) \leftrightarrow r$ — (true) — false — unknown

Question 2: Some Simple Tasks…

Show your calculations for each task.

a) Use the prime factorization method to determine gcd(5000, 600).

$5000 = 2^3 \cdot 5^4$
$600 = 2^3 \cdot 3^1 \cdot 5^2$
Then $\gcd(5000, 600) = 2^3 \cdot 3^0 \cdot 5^2 = 200$

b) Use the prime factorization method to determine lcm(30, 42).

$30 = 2^1 \cdot 3^1 \cdot 5^1$
$42 = 2^1 \cdot 3^1 \cdot 7^1$
Then $\mathrm{lcm}(30, 42) = 2^1 \cdot 3^1 \cdot 5^1 \cdot 7^1 = 210$

c) Convert the integer $(1101001)_2$ from binary expansion to decimal expansion.

$(1101001)_2 = 2^6+2^5+2^3+2^0 = 64 + 32 + 8 + 1 = 105$

d) Convert the integer 5322 from decimal expansion to hexadecimal expansion.

5322 / 16 = 332 R 10
332 / 16 = 20 R 12
20 / 16 = 1 R 4
1 / 16 = 0 R 1

Result: $(14CA)_{16}$

Question 3: Rules of Inference

Use rules of inference to show that the arguments below are valid, i.e., that their conclusion follows from their hypotheses. Apply the step-by-step method we used in class and list all those steps in your answer.

a) Hypotheses: If there is gas in the car, then I will go to the store. If I go to the store, then I will get a soda. There is gas in the car. Conclusion: I will get a soda.

g: "there is gas in the car" s: "I go to the store"
d: "I will get a soda"

Hypotheses:
$g \rightarrow s$

$s \rightarrow d$
g

Conclusion:
d

Step 1:	g	Hypothesis
Step 2:	$g \rightarrow s$	Hypothesis
Step 3:	s	R.I. using Steps 1 and 2
Step 4:	$s \rightarrow d$	Hypothesis
Step 5:	d	R.I. using Steps 3 and 4

b) Hypotheses: When Prof. P. gets angry, he fails his entire class. When the entire class fails, the Chancellor gets complaints. When the Chancellor gets complaints, he will either fire Prof. P., cut his salary, or do both. Prof. P. got angry, and he was not fired. Conclusion: Prof. P.'s salary was cut.

a: "Prof. P. gets angry"
f: "Prof. P. fails the entire class"
c: "Chancellor gets complaints"
p: "Chancellor fires Prof. P."
s: "Chancellor cuts Prof. P.'s salary"

Hypotheses:
$a \rightarrow f$
$f \rightarrow c$
$c \rightarrow (p \vee s)$
a
$\neg p$

Conclusion:
s

Step 1:	a	Hypothesis
Step 2:	$a \rightarrow f$	Hypothesis
Step 3:	f	R.I. Steps 1 and 2
Step 4:	$f \rightarrow c$	Hypothesis
Step 5:	c	R.I. Steps 3 and 4
Step 6:	$c \rightarrow (p \vee s)$	Hypothesis
Step 7:	$p \vee s$	R.I. Steps 5 and 6
Step 8:	$\neg p$	Hypothesis
Step 9:	s	R.I. Steps 7 and 8

Question 4: Complexity

Take a look at the following algorithm written in pseudocode:

```
procedure mystery(a1, a2, …, an: integer)
i := 1
while (i < n and ai ≤ ai+1)
  i := i + 1
if i = n then print "Yes!"
else print "No!"
```

a) What property of the input sequence $\{a_n\}$ does this algorithm test?

It tests whether the elements of the sequence are sorted in ascending order.

b) What is the computational complexity of this algorithm, i.e., the number of comparisons being computed as a function of the input size *n*?

Let us consider the worst case again. It occurs when the list is sorted, because then all pairs of consecutive elements have to be compared. Then for a given n, we iterate through the loop (n – 1) times and each time perform two comparisons, for a total of (2n – 2) comparisons.

When we reach the `while` statement for the n-th time, we only perform the comparison $i < n$, which yields false, and therefore, the comparison following the `and` is not executed. Therefore, we have performed (2n – 1) comparisons so far.

Finally, we need one more comparison to check whether the output should be "yes!" or "No!", which means that we made a total number of comparisons f(n) = 2n.

c) Provide a reasonable upper bound for the growth of the complexity function by using the big-O notation (no proof necessary).

Well, 2n is a polynomial of degree 1, which means that the complexity is O(n).

Question 5 (Bonus): Functions

Explain in your own words (no numbers or equations are necessary) why only bijective functions have an inverse function.

In order to be bijective, a function has to be injective and surjective. Let us show that whenever a function lacks one of these properties, it cannot have an inverse.

If a function f is not injective, it means that there are two different inputs x and y such that f(x) = z and f(y) = z. That means the inverse function f^{-1} would have to be defined such that $f^{-1}(z) = x$ and $f^{-1}(z) = y$, and thus, by definition, it would not be a function.
If a function is not surjective, it means that there is at least one value z in the codomain such that there is no x in the domain with f(x) = z. Therefore, $f^{-1}(z)$ would be undefined for z, which is in the codomain of f and should thus be in the domain of f^{-1}. However, a function needs to be defined for all inputs from its domain, and therefore, this function f^{-1} does not exist.