Your Name: Mitesh Patel Your UIN: 124002210

CSCE 222

CORRECTED MIDTERM I

Fall 2015

Problem [1]

Truth table for $\neg(p \vee \neg q) \wedge (r \to q)$

(1)

p	q	r	$\neg q$	$p \vee \neg q$	$\neg(p \vee \neg q)$	$r \to q$	$\neg(p \vee \neg q) \wedge (r \to q)$
T	T	T	F	T	F	T	F
T	T	F	F	T	F	T	F
T	F	F	T	T	F	T	F
T	F	T	T	T	F	F	F
F	F	F	T	T	F	T	F
F	T	F	F	F	T	T	T
F	F	T	T	T	F	F	F
F	T	F	F	F	T	T	T

(2) $p \vee q$ and $p \to q$ using only $\wedge$ and $\neg$.

$p \vee q \equiv \neg(\neg(p \vee q))$ using double negation law: $\neg\neg p \equiv p$

$\equiv \boxed{\neg(\neg p \wedge \neg q)}$ using demorgans law:

$\neg(p \vee q) \equiv \neg p \wedge \neg q$

$p \to q \equiv \neg(\neg(p \to q))$ using double negation law: $\neg\neg p \equiv p$

$\equiv \boxed{\neg(p \wedge \neg q)}$ using table 7: $\neg(p \to q) \equiv p \wedge \neg q$

(from book

(3) friendly - F

tall - T

not angry - $\neg A$

All friendly tall people are not angry

must be friendly and tall therefore use of $\wedge$ operator.

$$\boxed{\forall x\ (F(x) \wedge T(x)) \to \neg A(x)}$$

Problem [2]

1. (1) If the array is sorted, then the runtime is $O(n^2 \log n)$ — $s \to r$
   (2) The array being sorted is necessary for the algorithm to use procedure P. — $s \leftarrow p$
   (3) The runtime is not $O(n^2 \log n)$ — $\neg r$

1) $s \to r$
2) $p \to s$ p implies s because s is necessary for p
3) $\neg r$

Modus Tollens: $\dfrac{\neg q \quad p \to q}{\therefore \neg p}$

4) $\dfrac{\neg r \quad s \to r}{\therefore \neg s}$ from 1, 3 by Modus Tollens

5) $\dfrac{\neg s \quad p \to s}{\therefore \boxed{\neg p}}$ from 2, 4 by Modus Tollens

Algorithm doesnt use procedure P

2. Prove "n^2 is even" is equivalent to "n is even"

p – n^2 is even
q – n is even

$p \leftrightarrow q$

Proof by contrapositive:

1) $p \to q$ $p \to q \equiv \neg q \to \neg p$

So: $\neg q \to \neg p$

Means: n is odd $\to$ n^2 is odd

– Suppose n is odd

$n = 2k+1$ for some $k \in \mathbb{Z}$

– Proof n^2 is odd

$n = 2k+1$ $n^2 = (2k+1)^2$

$(2k+1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1$

$L = 2k^2 + 2k$

$= 2(L) + 1$ $L \in \mathbb{Z}$

Direct Proof

2) $q \to p$

Suppose n is even

– Proof n^2 is even

$n = 2k$ $k \in \mathbb{Z}$

$n^2 = 4k^2 : 2(2k^2) = 2(L)$ $L \in \mathbb{Z}$

Problem [3]

Prove that:

$\overline{A \cup B} = \bar{A} \cap \bar{B}$

we have to show that each set is a subset of the other

$\overline{A \cup B} \subseteq \bar{A} \cap \bar{B}$

show $(x \in \overline{A \cup B}) \rightarrow (x \in \bar{A} \cap \bar{B})$

Suppose 1) $x \in \overline{A \cup B}$

2) $x \notin A \cup B$ by complement

3) $\neg((x \in A) \vee (x \in B))$ by definition of intersection

4) $\neg(x \in A) \wedge \neg(x \in B)$ by demorgans law

5) $x \notin A \wedge x \notin B$ negation of propositions

6) $x \in \bar{A} \wedge x \in \bar{B}$ by complement

7) $x \in \bar{A} \cap \bar{B}$ by definiton of union we shown that $\overline{A \cup B} \subseteq \bar{A} \cap \bar{B}$

$\bar{A} \cap \bar{B} \subseteq \overline{A \cup B}$

show $(x \in \bar{A} \cap \bar{B}) \rightarrow (x \in \overline{A \cup B})$

Suppose 1) $x \in \bar{A} \cap \bar{B}$

2) $x \in \bar{A} \wedge x \in \bar{B}$ by definition union

3) $x \notin A \wedge x \notin B$ by complement

4) $\neg(x \in A) \wedge \neg(x \in B)$

5) $\neg((x \in A) \vee (x \in B))$ by demorgans law

6) $x \notin A \cup B$ by intersection

7) $x \in \overline{A \cup B}$ by complement we shown that $\bar{A} \cap \bar{B} \subseteq \overline{A \cup B}$

Because we have shown that each set is a subset of the other, the two sets are equal, and the identity is proven.

Problem [4]

Input : $a_1, a_2, \ldots a_n$: integers
Output : the k-th largest element of the list

```
for i=1 to k do ···· O(k)
    maxIndex = i  ·············· O(1)
    maxValue = a_i ············· O(1)
    for j=i+1 to n do ···· O(n)
        if a_j > MaxValue then ···· O(1)
            maxIndex = j ········ O(1)
            maxValue = a_j ······· O(1)
        end
    end
    swap a_i and a_maxindex ······ O(1)
end
return a_k
```

- iterates through inner loop n times on first iteration of outer loop, then n times on second and etc...

inner loop ;

$$O(n) \cdot (\overbrace{O(1) + O(1) + O(1)}^{O(1)})$$

$$= O(n) \cdot O(1) = O(n) \quad // \text{ executes } n \text{ times}$$

outer loop ;

$$O(k) \cdot (\overbrace{O(1) + O(1) + O(1)}^{O(1)} + O(n) \cdot (\overbrace{O(1) + O(1) + O(1)}^{O(1)}))$$

$$= O(k) \cdot (O(1) + O(n) \cdot O(1))$$

$$= O(k) \cdot O(n)$$

$$= \boxed{O(nk)}$$

Problem [5]

Prove by induction n that

$$\sum_{i=0}^{n-1} i2^i = 2 + (n-2)2^n$$

$$P(n) = \sum_{i=0}^{n-1} i2^i = 2 + (n-2)2^n$$

Basis: $P(1) = \sum_{i=0}^{0} i2^i = 0 = 2 + (1-2)2^1$

$= 2 + (-2) = 0$ ✓ Basis step true.

Inductive: $P(K) \to P(K+1)$

- Assume: $P(K) = \sum_{i=0}^{K-1} i2^i = 2 + (K-2)2^K$

- Show: $P(K+1) = \sum_{i=0}^{K} i2^i = 2 + (K-1)2^{K+1}$

$$\sum_{i=0}^{K} i2^i = K2^K + \sum_{i=0}^{K-1} i2^i \leftarrow \text{using IH}$$

$= K2^K + 2 + (K-2)2^K$

$= K2^K + 2 + K2^K - 2^{K+1}$

$= 2K2^K + 2 - 2^{K+1}$

$= K2^{K+1} + 2 - 2^{K+1}$

$= 2^{K+1}(K-1) + 2$ So by PMI it is Valid.

Bonus:

Recursive definitions for:

$l(T)$:= # of leaves of the full binary tree T, and

$i(T)$:= # of internal nodes of the full binary Tree T.

$l(T)$:

Basis: The root r is a leaf of the full binary tree w/ exactly one vertex r. This tree has no internal vertices. $l(r) = 1$ ✓

Recursive: Suppose T_1 and T_2 are full binary trees. Define new tree $l(T)$ that consist of nodes T_1 and T_2 attached. then $l(T) = l(T_1) + l(T_2)$

$i(T)$:

Basis: the root r is a leaf of the full binary tree w/ exactly one vertex r. this tree has, no internal node. $i(r) = \phi$ ✓

Recursive: Suppose T_1 and T_2 are full binary trees. Then the number of nodes of the full binary tree $T = T_1 \cdot T_2$ is $i(T) = 1 + i(T_1) + i(T_2)$