Assignment 1 – Discrete Review

Answer the questions in the boxes provided on the question sheets. If you run out of room for an answer, add a page to the end of the document.

Related Readings: `http://pages.cs.wisc.edu/~hasti/cs240/readings/`

Name: Yongsang Park

Wisc id: 908 022 3069

Logic

1. Using a truth table, show the equivalence of the following statements.

   (a) $P \vee (\neg P \wedge Q) \equiv P \vee Q$

P	Q	$\neg P$	$\neg P \wedge Q$	$P \vee (\neg P \wedge Q)$	$P \vee Q$
T	T	F	F	T	T
T	F	F	F	T	T
F	T	T	T	T	T
F	F	T	F	F	F

$\Rightarrow P \vee (\neg P \wedge Q) \equiv P \vee Q$ $\blacksquare$

   (b) $\neg P \vee \neg Q \equiv \neg(P \wedge Q)$

P	Q	$\neg P$	$\neg Q$	$\neg P \vee \neg Q$	$P \wedge Q$	$\neg(P \wedge Q)$
T	T	F	F	F	T	F
T	F	F	T	T	F	T
F	T	T	F	T	F	T
F	F	T	T	T	F	T

$\Rightarrow \neg P \vee \neg Q \equiv \neg(P \wedge Q)$ $\blacksquare$

(c) $\neg P \vee P \equiv \text{true}$

P	$\neg P$	$\neg P \vee P$
T	F	T
F	T	T

$\Rightarrow \neg P \vee P \equiv \text{true}$ $\blacksquare$

(d) $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

P	Q	R	$Q \wedge R$	$P \vee (Q \wedge R)$	$P \vee Q$	$P \vee R$	$(P \vee Q) \wedge (P \vee R)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	F	T	F	F
F	F	T	F	F	F	T	F
F	F	F	F	F	F	F	F

$\Rightarrow P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$ $\blacksquare$

Sets

2. Based on the definitions of the sets A and B, calculate the following: $|A|$, $|B|$, $A \cup B$, $A \cap B$, $A \setminus B$, $B \setminus A$.

(a) $A = \{1, 2, 6, 10\}$ and $B = \{2, 4, 9, 10\}$

$|A| = 4$
$|B| = 4$
$A \cup B = \{1, 2, 4, 6, 9, 10\}$
$A \cap B = \{2, 10\}$
$A \setminus B = \{1, 6\}$
$B \setminus A = \{4, 9\}$

(b) $A = \{x \mid x \in \mathbb{N}\}$ and $B = \{x \in \mathbb{N} \mid x \text{ is even}\}$

$|A| = \infty$
$|B| = \infty$
$A \cup B = A$
$A \cap B = B$
$A \setminus B = \{x \in \mathbb{N} \mid x \text{ is odd}\}$
$B \setminus A = \emptyset$

Relations and Functions

3. For each of the following relations, indicate if it is reflexive, antireflexive, symmetric, antisymmetric, or transitive.

(a) $\{(x, y) : x \leq y\}$

reflexive, antisymmetric, transitive

(b) $\{(x, y) : x > y\}$

antireflexive, antisymmetric, transitive

(c) $\{(x, y) : x < y\}$

antireflexive, antisymmetric, transitive

(d) $\{(x, y) : x = y\}$

reflexive, symmetric, transitive

4. For each of the following functions (assume that they are all $f : \mathbb{Z} \to \mathbb{Z}$), indicate if it is surjective (onto), injective (one-to-one), or bijective.

(a) $f(x) = x$

bijective

(b) $f(x) = 2x - 3$

injective

(c) $f(x) = x^2$

surjective

5. Show that $h(x) = g(f(x))$ is a bijection if $g(x)$ and $f(x)$ are bijections.

$g(x)$: one-to-one & onto
$f(x)$: one-to-one & onto

$g(f(x)) = g(f(y))$
$\Rightarrow f(x) = f(y)$ (Since g is one-to-one)
$\Rightarrow x = y$ (Since f is one-to-one)
$\Rightarrow g(f(x))$ is one-to-one

let $\forall a \in g(x)$
Since g is onto, there exists b such that $g(b) = a$
and also for $\forall x \in f(x)$, $f(x) = b$ exists.
$g(f(x)) = g(b) = a$
$\Rightarrow g(f(x))$ is onto

$\Rightarrow h(x)$ is bijective.

Induction

6. Prove the following by induction.

(a) $\sum_{i=1}^{n} i = n(n+1)/2$

$P(n) = \sum_{i=1}^{n} i = n(n+1)/2$

Basecase. $P(1)$ $\quad \sum_{i=1}^{1} i = 1$ $\quad$ true

$1 \cdot 2/2 = 1$

Inductive case

assume $P(k)$, or $\sum_{i=1}^{k} i = k(k+1)/2$ is correct for $\forall k \in \mathbb{N}$. I will show $P(k+1)$ is also correct.

$\frac{(k+1)(k+2)}{2} = \frac{(k+1)k + 2(k+1)}{2} = \frac{(k+1)k}{2} + (k+1) = \sum_{i=1}^{k} i + (k+1)$

$P(k+1)$ holds when $P(k)$ is correct. ∎

(b) $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$

$P(n) = \sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$

Basecase. $P(1)$ $\quad \sum_{i=1}^{1} i^2 = 1$ $\quad 1 \cdot 2 \cdot 3/6 = 1$ $\quad$ true

Inductive case

assume $P(k)$, or $\sum_{i=1}^{k} i^2 = k(k+1)(2k+1)/6$ is correct for $\forall k \in \mathbb{N}$. I'll show that $P(k+1)$ holds.

$1^2 + 2^2 + \cdots + k^2 + (k+1)^2 = k(k+1)(2k+1)/6 + k^2 + 2k + 1 = \frac{2k^3 + 3k^2 + k + 6k^2 + 12k + 6}{6} = \frac{2k^3 + 9k^2 + 13k + 6}{6} = \frac{(k+1)(k+2)(2k+3)}{6}$

$\Rightarrow$ $P(k+1)$ holds when $P(k)$ is true. ∎

(c) $\sum_{i=1}^{n} i^3 = n^2(n+1)^2/4$

$P(n) = \sum_{i=1}^{n} i^3 = n^2(n+1)^2/4$

Basecase $P(1)$ $\quad \sum_{i=1}^{1} i^3 = 1$ $\quad 1 \cdot 4/4 = 1$ $\quad$ true

Inductive case

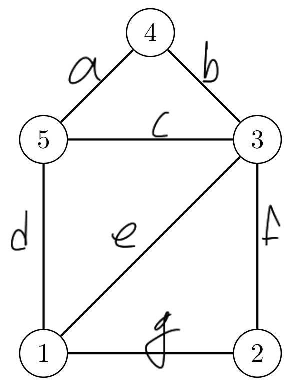
assume $P(k)$, or $\sum_{i=1}^{k} i^3 = k^2(k+1)^2/4$ is true for $\forall k \in \mathbb{N}$. I'll show $P(k+1)$ holds.

$\sum_{i=1}^{k+1} i^3 = 1 + \cdots + k^3 + (k+1)^3 = \frac{k^2(k+1)^2}{4} + (k+1)^3 = \frac{k^2(k+1)^2 + 4(k+1)^3}{4} = \cdots = \frac{(k+1)^2(k+2)^2}{4}$

$\Rightarrow$ when $P(k)$ is true, $P(k+1)$ holds. ∎

Graphs and Trees

7. Give the adjacency matrix, adjacency list, edge list, and incidence matrix for the following graph.



adjacency matrix

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

adjacency list

1 → 2 → 3 → 5
2 → 1 → 3
3 → 1 → 2 → 4 → 5
4 → 3 → 5
5 → 1 → 3 → 4

edge list

[1,2], [1,3], [1,5], [2,3], [3,4], [3,5], [4,5]

incidence matrix

	1	2	3	4	5
a	0	0	0	1	1
b	0	0	1	1	0
c	0	0	1	0	1
d	1	0	0	0	1
e	1	0	1	0	0
f	0	1	1	0	0
g	1	1	0	0	0

8. How many edges are there is a complete graph of size n? Prove by induction.

$p(n) = \sum_{i=0}^{n-1} i = \frac{n(n-1)}{2}$

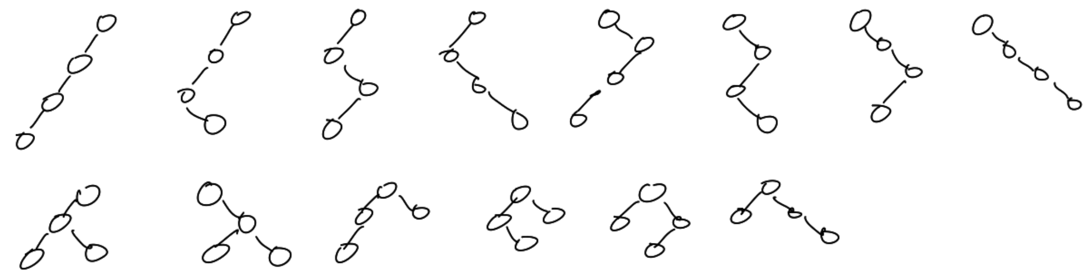
Basecase $P(1)$ $\sum_{i=0}^{0} i = 0$. $\frac{1 \cdot 0}{2} = 0$ true

Inductive case assume $P(n)$ holds for $\forall n \in \mathbb{N}$. I'll show $P(n+1)$ also holds.

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

$P(n+1)$ is also true when $P(n)$ is true.

9. Draw all possible (unlabelled) trees with 4 nodes.



10. Show by induction that, for all trees, $|E| = |V| - 1$.

Basecase $n=1 \Rightarrow V=1, E=0$ (single node)

when $n \geq 1$, the tree has n nodes & $n-1$ edges.

$\Rightarrow P_V(n) = n,\ P_E(n) = n-1$

Assume $P_V(k)$ & $P_E(k)$ holds for $\forall k \in \mathbb{N}$

Given that $P_V(k)$ & $P_E(k)$ are true, $P_V(k+1)$ & $P_E(k+1)$ are also true.

Counting

11. How many 3 digit pin codes are there?

— — — $10^3 = 1000$

12. What is the expression for the sum of the ith line (indexing starts at 1) of the following:

1
2 3
4 5 6
7 8 9 10
⋮

$$\sum_{k=1}^{\frac{i(i+1)}{2}} k - \sum_{k=1}^{\frac{(i-1)i}{2}} k = \frac{\frac{i(i+1)}{2}\cdot\left(\frac{i(i+1)}{2}+1\right)}{2} - \frac{\frac{(i-1)i}{2}\cdot\left(\frac{(i-1)i}{2}+1\right)}{2} = \frac{i}{4}\left((i+1)\cdot\frac{i^2+i+2}{2} - (i-1)\cdot\frac{i^2-i+2}{2}\right)$$

$$= \frac{i}{8}\left(i^3+2i^2+3i+2 - \left(i^3-2i^2+3i-2\right)\right) = \frac{i}{8}\left(4i^2+4\right) = \boxed{\frac{i(i^2+1)}{2}}$$

13. A standard deck of 52 cards has 4 suits, and each suit has card number 1 (ace) to 10, a jack, a queen, and a king. A standard poker hand has 5 cards. For the following, how many ways can the described had be drawn from a standard deck.

(a) A royal flush: all 5 cards have the same suit and are 10, jack, queen, king, ace.

4

(b) A straight flush: all 5 cards have the same suit and are in sequence, but not a royal flush.

$9 \times 4 = 36$

(c) A flush: all 5 cards have the same suit, but not a royal or straight flush.

$4 \cdot {}_{13}C_5 - 40 = 4 \cdot \frac{13 \cdot 12 \cdot 11 \cdot 10 \cdot 9}{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} - 40 = 5148 - 40 = 5108$

(d) Only one pair (2 of the 5 cards have the same number/rank, while the remaining 3 cards all have different numbers/ranks):

${}_{13}C_1 \cdot {}_4C_2 \cdot {}_{12}C_3 \cdot ({}_4C_1)^3 = 1098240$

Proofs

14. Show that $2x$ is even for all $x \in \mathbb{N}$.
    (a) By direct proof.

By the definition of an even number, $\forall y \in N$ is even if $y=2x$ for $\forall x \in N$.

(b) By contradiction.

Assume $2x$ is odd. for $k \in N$

$2x = 2k+1$

$2x-2k=1$

$x-k=\frac{1}{2}$

Because $x \in N$ & $k \in N$, they cannot make $\frac{1}{2}$ when subtracting.

$\Rightarrow$ $2x$ is even for $\forall x \in N$

15. For all $x, y \in \mathbb{R}$, show that $|x+y| \leq |x| + |y|$. (Hint: use proof by cases.)

1) $x \geq 0, y \geq 0 \rightarrow |x+y| = |x|+|y|$
2) $x<0, y \geq 0 \rightarrow |x+y| < |x|+|y|$
3) $x \geq 0, y<0 \rightarrow |x+y| < |x|+|y|$
4) $x<0, y<0 \rightarrow |x+y| = |x|+|y|$

$\Rightarrow |x+y| \leq |x|+|y|$ is true

Program Correctness (and invariants)

16. For the following algorithms, describe the loop invariant(s) and prove that they are sound and complete.

(a)

Algorithm 1: findMin

Input: a: A non-empty array of integers (indexed starting at 1)
Output: The smallest element in the array

```
begin
    min ← ∞
    for i ← 1 to len(a) do
        if a[i] < min then
            min ← a[i]
        end
    end
    return min
end
```

Soundness – the algorithm yields true result (partial correctness)
Completeness – the algorithm terminates for all valid input (termination)

Assume valid input "non-empty array of integers" for which the program will terminate.
We need to show "min = the smallest element in the array".
By for loop, it visits every element in the input array.
if an element in the input array < min, min ← element.
When exiting the for loop, min indicates the smallest element in the array.
∴ When findMin returns min, it contains the correct value.

Sound

Assume the valid input "non-empty array".
The value i is from 1 to the length of the array according to the for-loop definition.
When the loop goes on, i is increased by 1 and it will reach to the length of the array eventually. And the code will be end.

Completeness

(b)

Algorithm 2: InsertionSort

Input: a: A non-empty array of integers (indexed starting at 1)
Output: a sorted from largest to smallest

```
begin
    for i ← 2 to len(a) do
        val ← a[i]
        for j ← 1 to i − 1 do
            if val > a[j] then
                shift a[j..i − 1] to a[j + 1..i]
                a[j] ← val
                break
            end
        end
    end
    return a
end
```

Assume a valid input "non-empty array of integers" for which the algorithm terminates.
We need to show "a= a sorted array from largest to smallest",
By the first loop, an indicator(i) will hold from index 2 to the end.
By the second loop, the other indicator(j) will hold from index 1 to the first indicator.
The variable 'val' is set as $a[i]$, and according to the if statement, if $a[i] > a[j]$,
$a[i]$ value and $a[j]$ value are exchanged.
It returns 'a = a sorted array from largest to smallest."

Sound

Assume the valid input "non-empty array of integers".
In the first for-loop, i starts from 2 to the length of the input array.
Every statement inside the loop ends, i increases and if $i >$ length, the loop ends.
In the inner for-loop, j starts from 1 to $(i-1)$. i starts from 2, so $1 \leq i-1 \leq (\text{length} - 1)$.
After the if-statement ends, j increases and when $j > i-1$, the loop ends,
After two loops end, the "insertionSort" ends.

Completeness

Recurrences

17. Solve the following recurrences.

(a) $c_0 = 1; c_n = c_{n-1} + 4$

$C_0 = 1$
$C_1 = 5$
$C_2 = 9$
$\vdots$
$\boxed{C_n = 4n + 1}$

(b) $d_0 = 4; d_n = 3 \cdot d_{n-1}$

$d_0 = 4$
$d_1 = 3 \cdot 4$
$d_2 = 3^2 \cdot 4$
$\vdots$
$\boxed{d_n = 3^n \cdot 4}$

(c) $T(1) = 1; T(n) = 2T(n/2) + n$ (An upper bound is sufficient.)

$$\frac{T(n)}{n} = \frac{T(n/2)}{\frac{n}{2}} + 1$$

$$T(n/2) = 2T(n/4) + \frac{n}{2}$$

$$\frac{T(n/2)}{n/2} = \frac{T(n/4)}{n/4} + 1$$

$$\Rightarrow \frac{T(2^k)}{2^k} = T(1) + k = k+1$$

$$\Rightarrow T(2^k) = 2^k(k+1)$$

if you put $k = \log_2 n$

$$\boxed{T(n) = n(\log_2 n + 1)}$$

(d) $f(1) = 1; f(n) = \sum_1^{n-1} (i \cdot f(i))$
(Hint: compute $f(n+1) - f(n)$ for $n > 1$)

$$f(n+1) - f(n) = \sum_1^{n}(i \cdot f(i)) - \sum_1^{n-1}(i \cdot f(i)) = n f(n)$$

$$\Rightarrow f(n+1) = (n+1) f(n)$$

$$\begin{aligned} f(n) &= n f(n-1) = n \cdot (n-1) f(n-2) \\ &= n(n-1)(n-2) f(n-3) \\ &= \cdots \\ &= n \cdot (n-1) \cdots 2 \cdot f(1) \\ &= n \cdot (n-1) \cdots 2 \cdot 1 \\ &= n! \end{aligned}$$

$$\boxed{f(n) = n!}$$

Coding Question

Most assignments will have a coding question. You can code in C, C++, C#, Java, or Python. You will submit a Makefile and a source code file.

Makefile: In the Makefile, there needs to be a build command and a run command. Below is a sample Makefile for a C++ program. You will find this Makefile in assignment details. Download the sample Makefile and edit it for your chosen programming language and code.

```
#Build commands to copy:
#Replace g++ -o HelloWorld HelloWord.cpp below with the appropriate command.
#Java:
#       javac source_file.java
#Python:
#       echo "Nothing to compile."
#C#:
#       mcs -out:exec_name source_file.cs
#C:
#       gcc -o exec_name source_file.c
#C++:
#       g++ -o exec_name source_file.cpp
#Rust:
#       rustc source_file.rs

build:
        g++ -o HelloWorld HelloWord.cpp

#Run commands to copy:
#Replace ./HelloWorld below with the appropriate command.
#Java:
#       java source_file
#Python 3:
#       python3 source_file.py
#C#:
#       mono exec_name
#C/C++:
#       ./exec_name
#Rust:
#       ./source_file

run:
        ./HelloWorld
```

HelloWorld Program Details The input will start with a positive integer, giving the number of instances that follow. For each instance, there will be a string. For each string s, the program should output Hello, s! on its own line.

A sample input is the following:

```
3
World
Marc
Owen
```

The output for the sample input should be the following:

```
Hello, World!
Hello, Marc!
Hello, Owen!
```