# CPT_S 515: Homework #3

*Instructor: Zhe Dang*

**Sheng Guan**

# Problem 1

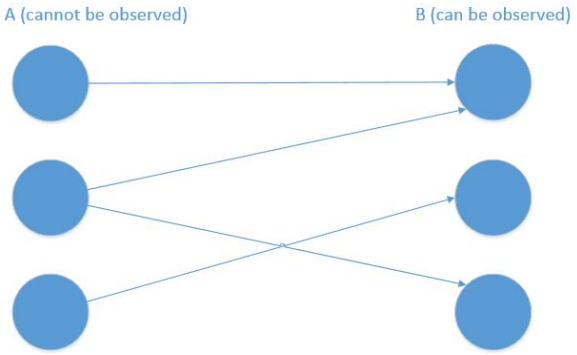
1. Let D be a device that keeps sending out messages. Each message contains two parts A and B where the intruder can not observe A but he can observe B. Suppose that each of A and B takes 10 bits so, each message is exactly 20 bits. Before the developers sell the device to the public (including the intruder), they run some experiments trying to make sure that there isnt any information leakage that is more than 1 bit from A to B in a message. The experiments are done by run the device for a long time and obtain a large and finite set C of messages. Please design a program that can estimate the average number of bits actually leaked from A to B in a message drawn from C.

**Answer:**

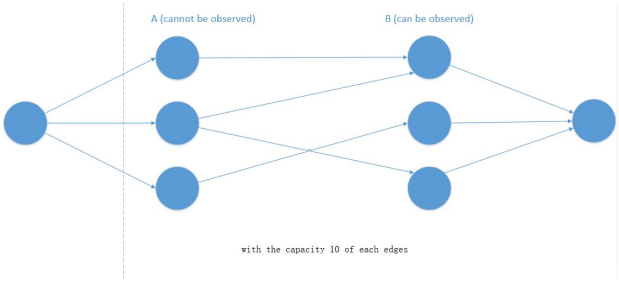
We first organize the useful information as below:

- 1.Nodes cannot be observed: :A, and nodes can be observed: B
- 2.A large finite set C of messages

We can assume the message sending repeat n times. n is a finite, large number. In each cycle, the C as $C_i$. Then, during each sending we can draw the connection between from A to B as following shows:




We can add a source and a sink to this graph. Then we can try to find the maximum flow.




The conclusion we made in class, the sum of information flow is: $log\mu$, $\mu$ is the sum of edges in the result of maximum flow.
The total practical transmission equals to $sum_{i=1}^{n} C_i$;
The total theoretical transmission equals to $nlog\mu$;
The difference value equals to $|nlog\mu - sum_{i=1}^{n} C_i|$;
Then, the average difference is $|log\mu - \frac{1}{n} sum_{i=1}^{n} C_i|$.
Thus, the average difference is $|log\mu - \frac{1}{n} C|$.

# Problem 2

2. Consider a C-function that has integer variables as arguments and integer as return type: int myFunction(int $x_1$, int $x_2$, ..., int $x_7$)
In the function with arguments $x_1, x_2, ..., x_7$ which are integer variables, there are only 10 lines of code, where each line is in the form of an assign- ment variable := Exp to an integer variable where Exp is a linear combi- nation of integer variables (e.g., $y := 2x1 + 3x2 - 5$) or an if-then-else statement where the condition is a comparison between two linear constraints on integer variables and the assignments in the if-then-else statement are in the form of variable := Exp shown above (e.g., if $(y > 12x_1 - z)$ then $x2 := 3x7 - 15$ else $x5 := 18x4 - 6x7 + 6$. The first line of the function declares three integer variable x,y,z, while the last line is to return the value x back. Please design a program that can verify whether there are values for $x_1, x_2, ..., x_7$ passed to the function that can make the function return a negative integer.
**Answer:**
There are ten lines of code, and there are two types of conditions:

- 1) If it is in the form like: $variable := Exp \rightarrow it$ is already an integer linear equation;
- 2) Or it is a comparison form of if-then-else, in the if or else condition, we still can have an integer linear equation.

The first line declares three integer variable x,y,z and the last one just returns x. Thus, we only consider eight linear equations in this case. For example, we will have:
$a_{11}x_1 + a_{12}x_2 + \cdots + a_{17}x_7 + b_1x + c_1y + d_1z = 0$
...
...
We can always put these integer linear equations in a standard matrix form.
In the end, we will get:

$$Ax = 0 \quad ,$$

where A is a 8-by-10 matrix and $x = (x_1x_2x_3 \dots x_7xyz)^T$
Now we have this linear programming problem and we can get min x subject to

$$Ax = 0, x \neq 0$$

We can solve this linear programming problem and get the min x.
If the optimal min x does not satisfy the returning x is negative, then we can get the conclusion that the program will verify true. Otherwise, the program will verify false.

# Problem 3

3. Symbolic representation is way to code a finite object. BDD is a way to code a finite set. However, when a power set (a set of finite sets) is given, BDD is not usually efficient. Sometimes, it is a good idea to code an object as a number since a number itself is a string (e.g., 123 is the string "123"). We now consider a special case. Let K =1,...,k for some k, and consider P be a set of disjoint subsets of K. That is, $P = K_1, ..., K_m$ for some m and each $K_i \subseteq K$ and $K_i \cap K_j = \emptyset$ whenever $i \neq j$. I want to design an algorithm, for a given K, to code (or transform or represent) each such P into a number $C_P$ such that the code C is optimal; i.e.,

- (1).C is 1-1,
- (2). $C_P \in 1, ..., B_K$,

where $B_K$ is the number of all such P's for the given K.

**Answer:**

As the problem indicates, the K =1,...,k.Then, let we group these k elements into the $P = K_1, ..., K_m$. The goal for this problem is to transform each such P into a number $C_p$, and the code C is 1-1 mapping.

We use an example to illustrate our idea here.

For example , we can let $K_1 = 1, 2, 3, K_2 = 4, 5, K_3 = 6, 7, K_4 = 8$, ... .etc. We pick one element k in the K and put it into $K_i (1 \leq i \leq m)$. Each k can be only picked once such that $K_i \cap K_j = \emptyset$. For we need to get a 1-1 mapping, so we need to map each element k into a vector.

The example vector can be shown below:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | k |
|---|---|---|---|---|---|---|---|---|---|---|
| K1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 |
| K2 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | ... | 0 |
| K3 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | ... | 0 |
| K4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ... | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| Km | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 |

Then each row can be represented as a binary-representation value and we will get m different binary-representation values. If we further transform these m binary-representation values as decimal-representation values 1,...,$B_k$. This will make the transform each such P into a number $C_p$.

# Problem 4

4. (easy) Let G be a directed graph of 2048 nodes. When we use a Boolean formula to represent the G, how many Boolean variables are needed in the formula?

**Answer:**

If there is a walk from $v_i$ to $v_j$ existing, we can draw the conclusion that it is reachable from $v_i$ to $v_j$. We use a Boolean variable $b_{ij}$ to describe whether it is reachable from $v_i$ to $v_j$ or not:

We can use a reachable matrix $R = \{b_{ij}\}_{2048*2048}$ to record.

We have $2^{11} = 2048$ and $2 * 11 = 22$ So, 22 Boolean variables are needed in the formula.

# Problem 5

5. (easy) Data types are an abstraction of data and data structures are a way to store the types into memory. In particular, we never store physical objects in memory; in case when we really want to do it, we first represent the physical objects in an abstract representation and store the representation in memory. Here is a problem. There are 40 students in a classroom. I want to design a closet so that any one of the students can be hidden inside. Imagine that the closet is a chunk of computer memory. Then, how big (in bits) closet do you need?

**Answer:**

We only store representations and use Boolean formula to represent them all. We have $2^5 = 32$ and $2^6 = 64$ Because $32 < 40 < 64$, we need to use 6 bits closet to store the representation.