EDA387 - Lab 4.1: Value Discovery in Complete Graphs

Nathan HAUDOT, Jinsong LI
September 14, 2021

1 Introduction

The goal of this exercise is to discover an unknown value in a complete graph. Each processor has two registers r_i and s_i. The last one being inaccessible for p_i, the goal of this exercise is to make the processor p_i find the value of its secret register with the help of the other processors. The r_i registers are shared in reading, and only writable by the corresponding p_i processor.

2 Assumptions

Each processor has two registers (r_i, s_i) described as follows:

- Each r_i register is readable by any p_i processor, with $0 < i < n$
- Each p_i processor can write in its own r_i register
- Each s_i register can not be read by its corresponding p_i processor
- For p_j processors such as $j \neq i$, s_i is accessible in reading
- We will assume that every processors knows n, defined as the total number of processors

3 Resolution

Step 1

For each p_i processor, we define r_i such as:

$$r_i = \sum_{\substack{j=0 \\ j \neq i}}^{n} s_j \tag{1}$$

Step 2

We set a variable named sum representing the sum of all the secret registers:

$$sum = \sum_{i=0}^{n} s_i \tag{2}$$

Now, we can assume that for every r_i with $0 < i < n$, $r_i = sum - s_i$. This is our first equation.

Step 3

The processors can sum all the registers together:

$$\begin{aligned} \sum_{j=0}^{n} r_j &= (sum - s_0) + (sum - s_1) + ... + (sum - s_n) \\ &= -(s_0 + s_1 + ... + s_n) + nsum \\ &= (n-1)sum \end{aligned} \tag{3}$$

Here, we assume that every p_i processor knows the value of n.

Step 4

Now we have two equations with the variable sum, which we can transform like this:

$$\begin{cases} \sum_{j=0}^{n} r_j = (n-1)sum \\ r_i = sum - s_i \end{cases} = \begin{cases} sum = \frac{1}{(n-1)} \sum_{j=0}^{n} r_j \\ sum = r_i + s_i \end{cases} \longrightarrow s_i = \frac{\sum_{\substack{j=0 \\ j \neq i}}^{n} r_j}{(n-1)} \tag{4}$$

4 Conclusion

Therefore, each p_i processor can compute the value of its own s_i register with the formula (4). However, this method has a constraint: we are obliged to identify each processor during the calculation.

By analyzing the method proposed in the discussions of Lab 4.1, we realized that the solution using XOR sums (and its properties) was much more powerful, and that we could retrieve the value of s_i using any p_x processor (with x different from i), as well as its r_x and s_x registers, which avoids having to identify the processors from each other.