# Assignment

Mohammad Khalid Udayagiri

Collaborative Text Editing

May 27, 2020

**The problem statement :** Given two users concurrently editing a shared document come up with a data model for storing the document such that the updates made by both the users are preserved and the final state of the document is the same.

# 1 Data Modeling

## 1.1 Storage and Structure of data

In a multi user collaborative environment, there are the following storage locations.

1. The local text editor
2. The local buffer

### 1.1.1 Local text editor

The data in the text editor will be stored as a linear sequence of alphabetic characters.

### 1.1.2 Local buffer data type

The buffer I am talking about is shared buffer across all users in the network. This buffer is the local replica of the data that every user has. Since the operations we are focusing on are Insert and Delete in a collaborative environment, the buffer data type should use CRDT to represent the data which will guarantee that Insert and Delete operations commute.

In this assignment I will be using **Treedoc** CRDT to store the data on the user's shared database. Treedoc data type allows users to insert a new atom and delete a new atom collaboratively into the shared data base. Treedoc supports the data to be updated such that the constraints given in the problem statement hold. As the name suggest treedoc allows users to store data in the form of a tree. Each node in the tree corresponds to a atom(in my case characters). The treedoc data type helps us in generating unique identifiers for each additional node. The identifier for a node is typically the labels concatenated from root the given node. When concurrent updates happen to the same node, the treedoc can be modified to sotore the labels of each node as (bit, siteId) where bit = 0 if the node is the left child of the parent, 1 otherwise and siteId is the identifier of the user who is trying to update the given node. This is the logical structure of the treedoc data type.

# 2 How is the data stored on the server?

I have Used UID's as keys and atom values as attributes to persist the treedoc in the data base. I have used the query number(primary key), timestamp, type of query, atom, position, site id to store the queries in the data base.

# 3 How is data presented to the user

Since each user has the shared buffer, we will just run an in-order traversal over the tree and concatenate all the values of the nodes visited. This array of characters will then be displayed by the text editor.

# 4 Estimates

## 4.1 Estimate for reading the research paper

I took me about 8-10 hrs time to completely understand the paper. I was done with the given paper by 23rd may.

## 4.2 Estimate for implementing

I took me about 5-6 hrs time to come up with first working prototype of treedoc in python, Since It has been a while I have coded in python, I expect the time to implement to go down with implementing more in python. The time taken for implementing final working prototype was about 10-12 hrs considering the time taken to learn to create and persist data in psql, other changes to the treedoc data type and processing queries.