NoDD Algorithm Implementation

Andrew Plum & Nathan Nguyen
Department of Computer Science, University of Idaho
plum0598@vandals.uidaho.edu
nguy10210@gmail.com

KEYWORDS

NoDD, Functional Dependency, Normalization, Evaluation Algorithm, and Project 360

ACM Reference Format:

Andrew Plum & Nathan Nguyen. 2023. NoDD Algorithm Implementation. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

NoDD (stands for Normalization and Database Design Theory) is an interactive assessment tool in development hosted on the Project 360 website. It exists to better teach students and check for their understanding of the concepts of functional dependencies and the process of normalizing them. Last semester's students designed and implemented a web interface. I, along with my project partner, intend to expand on the project from last semester by implementing one of the two functional dependency normalization algorithms for the purpose of creating a comprehensive evaluation system. This will give insight into the comprehension of the student with the subject matter.

2 BODY

The core of this project involves the integration of a normalization algorithm into an existing web interface. The integration enables us to evaluate the responses of the students in real time. There are several steps I must take to successfully implement this project.

The first phase of the project entails connecting with the web interface. I need to familiarize myself with the existing framework so that I understand the structure and functionality of the data. Afterward, I must collect relevant data and structure it so that it can be used in the algorithm in the next stage. This may involve parsing it so what is relevant can be collected.

The next step is the actual implementation of the algorithm. Regarding the schedule of the project, I intend to spend the majority of the time here as I consider this to be the heart of the project. How I intend to start this phase is with the development of a prototype algorithm that covers only a select number of cases. This way I can verify with an expected output to see if all of the underlying fundamentals of the code are functioning as intended before dedicating time to the creation of an algorithm that might have innate flaws in its overall design. After developing a successful prototype, I will then proceed by expanding upon it to cover all cases. To cover all cases, I will use both of the existing algorithms for the normalization of functional dependencies: Third Normal Form (3NF) and Boyce Codd Normal Form (BCNF). I will need to familiarize myself with these algorithms before I definitively decide which one I will implement in my program.

To implement the evaluation and logic algorithms, I will be using the scripting language Python, and potentially PHP and JavaScript to join with the existing NoDD program. The development tools I will use for the project include Visual Studio Code or Notepad++ as code editors, Git/GitHub for version control, and the pre-existing tools being used to host NoDD on Project 360. This is not a comprehensive list, but I expect these are the tools I will use for the project.

After the algorithm is implemented, I can normalize the initial functional dependency inputted by the student which should now give a correct output. This correct output can then be used to compare it with the student's response for the normal form of the functional dependency. This comparison should be enough to evaluate their response as correct or not and inform them as such. If time permits, I will implement functionality that informs the student of how correct their answer is if it is not completely correct, and also if the steps they took to normalize the functional dependency were concise in that they were the same as the steps outlined by the efficient algorithm implemented. Interactive tutoring messages would be implemented here to aid the students in helping them understand where in the process they were wrong.

There is a schedule I have outlined for the completion of the project. I expect familiarizing myself and interacting with the web interface to take the least amount of time with only

5 days allotted. Next, I have 21 days set for the creation of the prototype. Afterward, I assigned 30 days to the expansion of the prototype where I implemented the algorithm which covers all cases. If everything is done correctly, testing should take only a moderate amount of time with 7 days devoted to it. That leaves 7 days for flexibility and for finishing touches. Although the schedule is subject to change, the timeline set is achievable with work spread between my partner and me. The Gantt chart, figure 1, featured here is a visual representation of the project timeline.

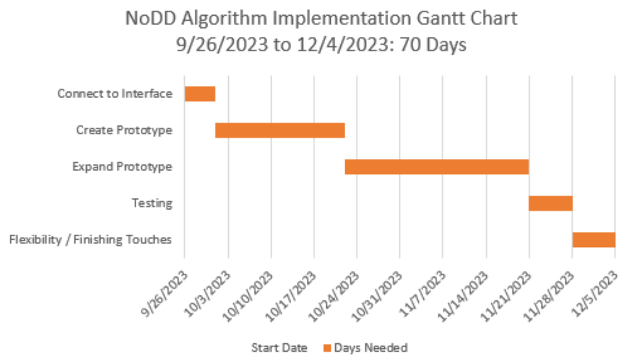

Figure 1: Project Timeline

REFERENCES

https://project360.smartdblab.org