

Modelling Students' Source Code Evolutions To Provide Personalised Guidance In Introductory Programming Courses

1 Background & Problem Statement

- The High demand for skilled professionals in Computing has led to large classrooms, which exhibit diversity in problem-solving approaches and learning behaviours.
- Learning Management Systems(LMS) collect valuable data that could offer insights into students' behaviour and interactions with programming tasks.
- However, this data is underutilized offering limited personalized guidance and leaving students stuck in a state of prolonged uncertainty during practical activities.

2 Aim

- Assist students' progress from stuck states in programming tasks
- Explore novel approaches that make use of Markov models in analysing the evolution of source codes

3 Research Design



Study 1: Development and Assessment of Model Accuracy

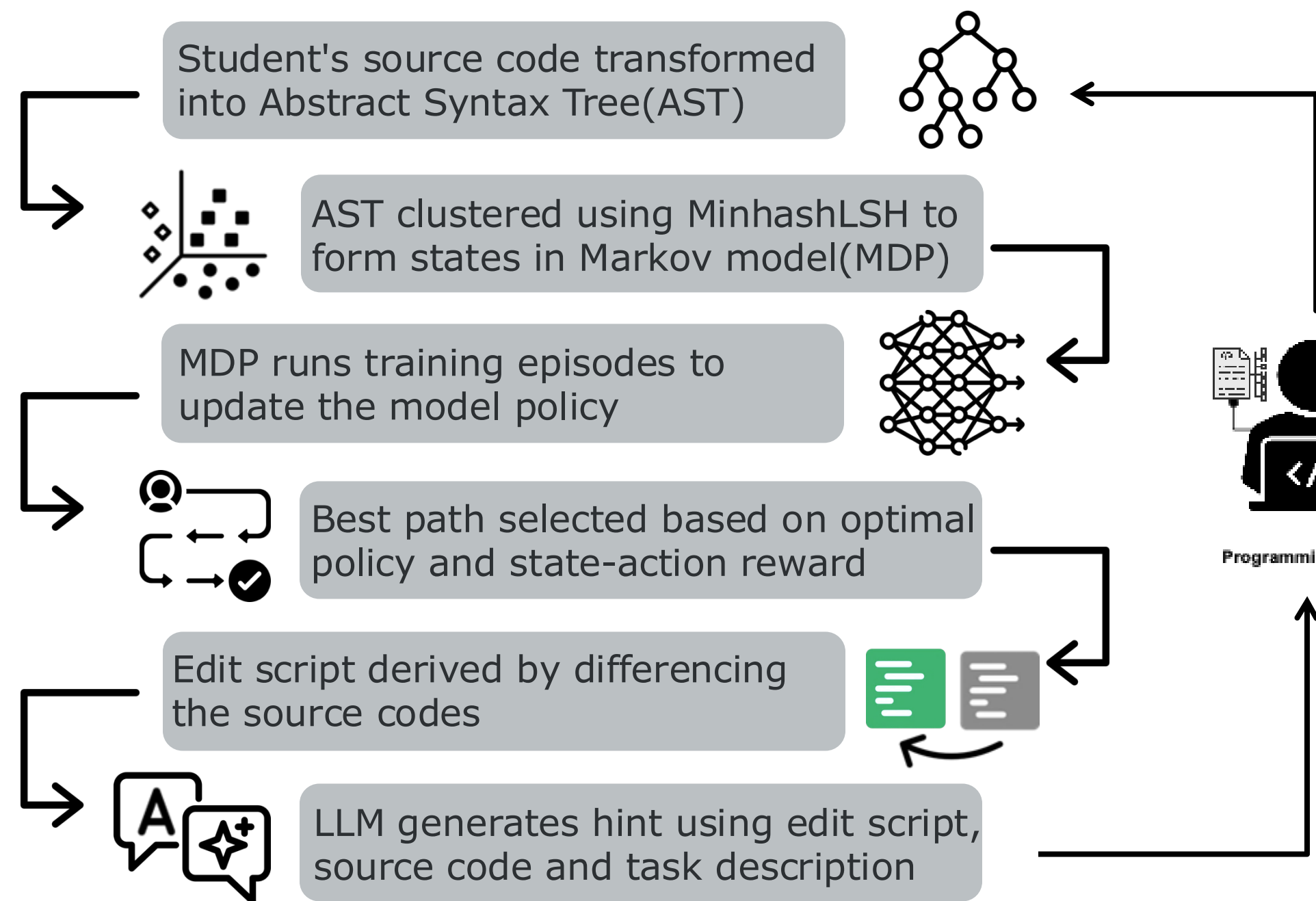


Study 2: Analysing Patterns in Students' Problem-Solving Behaviours Using Markov Models Visualised as FSMs

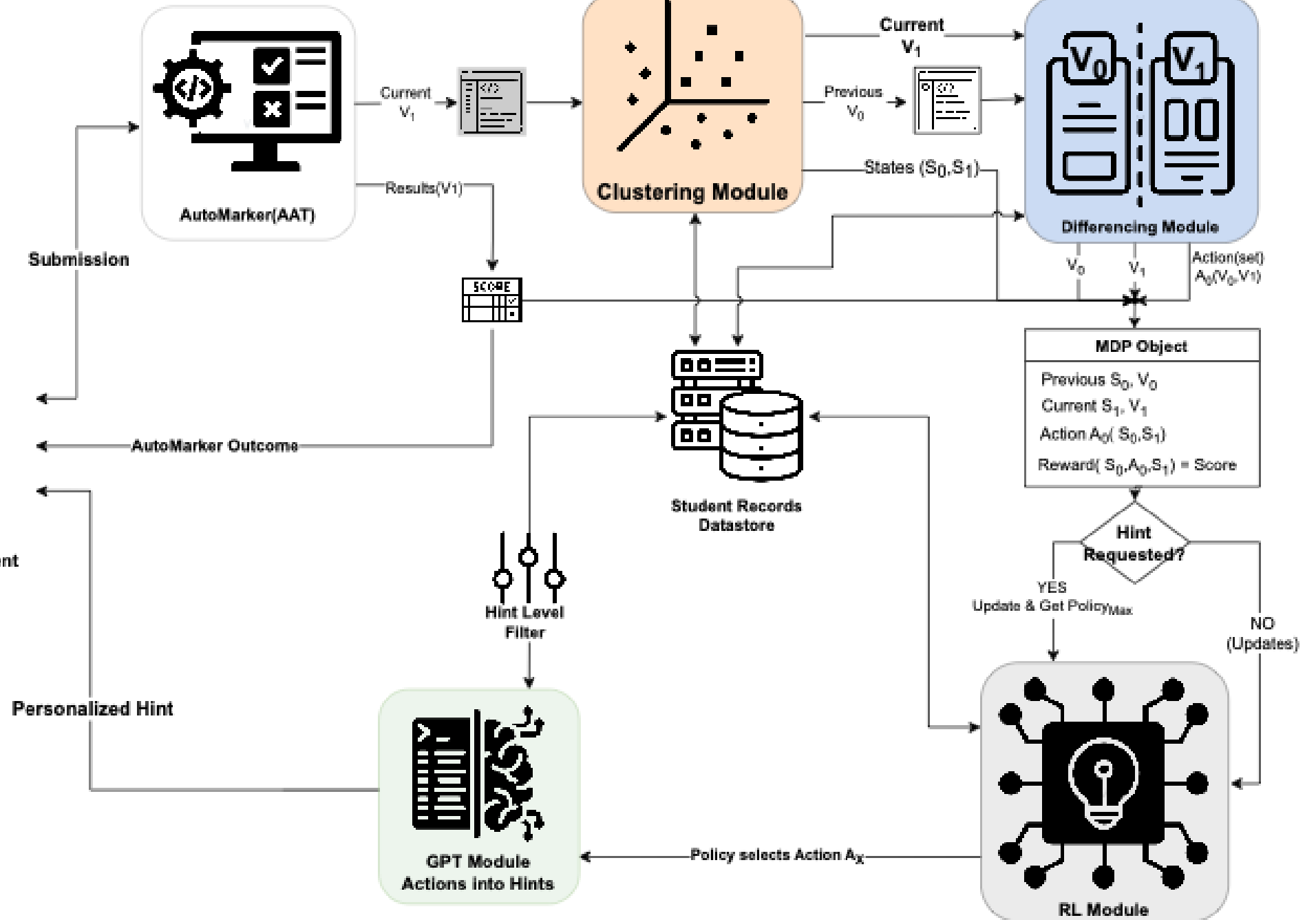


Study 3: Implementing and Assessing the Impact of the Personalised Guidance Tool on Programming Processes

4 Implementation - Process flow

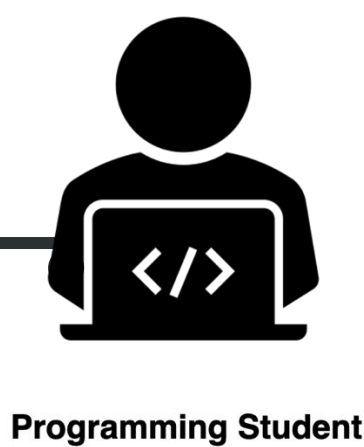


System Architecture



5 Results : Sample hints generated

```
main.py +
1 var1=input('Enter a sentence:\n')
2 var2=''
3 for i in range(0,len(var1)):
4     var2+=var1[i].lower()
5 var1=var2
6 var2,var3 = ''
7 var4='T'
8 for i in range(0,len(var1)):
9     if var4=='T':
15     elif var1[i]==' ':
24     elif i==len(var1)-1:
33     else:
35 print(var2)
```



3 Precise technical hints

- You've done well with the basic structure. Here are specific improvements to make:
- Convert the input to lowercase immediately after receiving it
 - Use 'find(" ")' to locate word boundaries in the sentence
 - Create a loop that processes one word at a time based on space positions
 - For each word, check if it starts with a vowel or consonant(s)
 - Apply the appropriate Pig Latin transformation:
 - Vowel-start: add "way" to the end
 - Single consonant: move it to the end and add "a" + consonant + "ay"
 - Multiple consonants: move them all to the end and add "a" + consonants + "ay"
 - Build the translated sentence by concatenating the transformed words with spaces

2 Generic conceptual hints

- You've got the right idea with case conversion and input handling. Here are some conceptual improvements:
- Consider processing the sentence word by word instead of character by character
 - Use string slicing to separate consonants from vowels at the start of each word
 - Implement different transformation rules based on whether a word starts with vowels or consonants
 - Track word positions using string methods like 'find()' to locate spaces between words
 - Store the translated words in a new string to build the final output

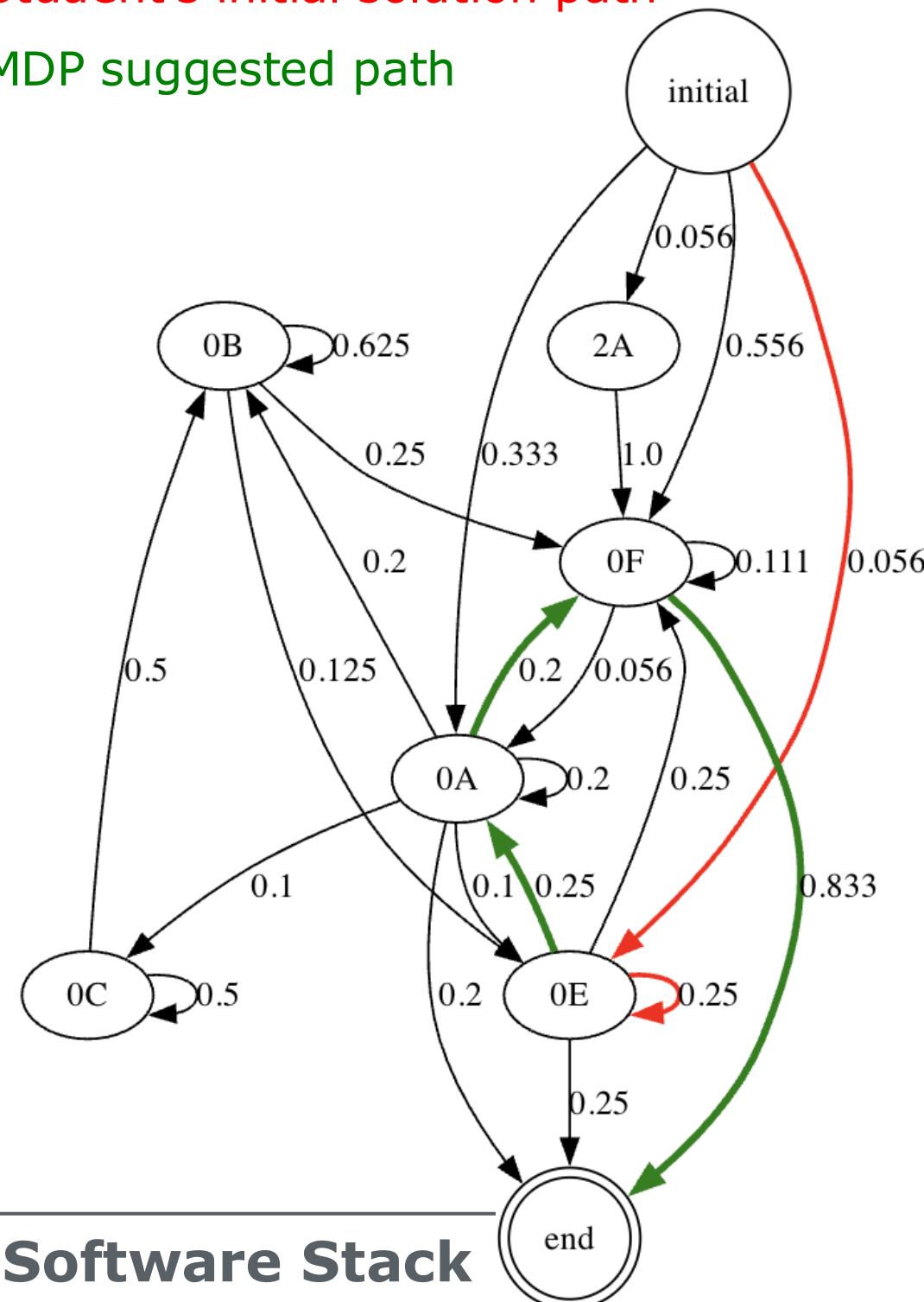
1 Inquiry based hints

Great job on starting the Pig Latin translator! You've correctly implemented the input handling and case conversion. Let's focus on improving the translation logic:

- How can we efficiently process each word in the sentence rather than character by character?
- What's a better way to track word boundaries than using a flag variable like 'var4'?
- How should we handle words that start with multiple consonants differently from those that start with a single consonant?
- Have you considered using string slicing to manipulate the words more effectively?

6 Solution Paths selected by MDP

- Student's initial solution path
- MDP suggested path



7 Software Stack



Designed with icons from Flaticon.com



UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD



Hasso Plattner Institut



HERMAN KANDJIMI

PhD Candidate | UCT | kndher001@myuct.ac.za

Supervisor: Prof. Hussein Suleman