Fontys University of Applied Sciences
School for ICT (FHICT)

Automata and Logic Engineering 1/2

ALE1/ALE2 Report

My Name

EMPTY version

Eindhoven, Month 202X

Abstract

[Draw your reader's attention with an engaging abstract. It is typically a short summary of the document.]

Preface

[If you want to, please write all your preface text here.]

Contents

List of Figures

Chapter 1

Introduction

[Start *general* and go towards *specific* information that your paper is based on. Generalizations that you anticipate readers are likely to accept can be effective opening sentences. By beginning with information that is generally accepted, you begin with something familiar to the readers, and ease them into your paper. Whether you begin an introduction text with a general statement or a definition is a matter of personal preference (Swales & Feak, 2012). However, sometimes one may be a strategically better choice than the other: think of your *target audience*. For instance, in a regular course is most likely that only the course instructor will be reading your paper, thus, your purpose is usually to display familiarity, expertise, and intelligence. In a graduation, there are several assessors involved which may possess less background, thus, your purpose is again to display familiarity, expertise, and intelligence, but through general insights.]

Chapter 2

Parse & Tree

[General notes:

- Describe your approach, detail upon challenges, what were your struggles, how you came across them.
- If you did not manage to implement, also mention all your attempts and why your solution could not be implemented.
- Describe (in existent) additional features or smart computations that I might miss while assessing your code.
- All team work (2+ students) must be mentioned; e.g.: "In class Tree.cs from line 15-55 is code developed together with partner X (PCN:xxxxxx)".
- IMPORTANT: Please make sure to mention all lines of code you worked together with another fellow student, otherwise plagiarism will be detected in your code and you will be sent to the examination board.]

Chapter 3

Truth table & Hash code

[General notes:

- Describe your approach, detail upon challenges, what were your struggles, how you came across them.
- If you did not manage to implement, also mention all your attempts and why your solution could not be implemented.
- Describe (in existent) additional features or smart computations that I might miss while assessing your code.
- All team work (2+ students) must be mentioned; e.g.: "In class Tree.cs from line 15-55 is code developed together with partner X (PCN:xxxxxx)".
- IMPORTANT: Please make sure to mention all lines of code you worked together with another fellow student, otherwise plagiarism will be detected in your code and you will be sent to the examination board.]

Chapter 4

Simplify

[General notes:

- Describe your approach, detail upon challenges, what were your struggles, how you came across them.
- If you did not manage to implement, also mention all your attempts and why your solution could not be implemented.
- Describe (in existent) additional features or smart computations that I might miss while assessing your code.
- All team work (2+ students) must be mentioned; e.g.: "In class Tree.cs from line 15-55 is code developed together with partner X (PCN:xxxxxx)".
- IMPORTANT: Please make sure to mention all lines of code you worked together with another fellow student, otherwise plagiarism will be detected in your code and you will be sent to the examination board.]

Chapter 5

Normalize

[General notes:

- Describe your approach, detail upon challenges, what were your struggles, how you came across them.
- If you did not manage to implement, also mention all your attempts and why your solution could not be implemented.
- Describe (in existent) additional features or smart computations that I might miss while assessing your code.
- All team work (2+ students) must be mentioned; e.g.: "In class Tree.cs from line 15-55 is code developed together with partner X (PCN:xxxxxx)".
- IMPORTANT: Please make sure to mention all lines of code you worked together with another fellow student, otherwise plagiarism will be detected in your code and you will be sent to the examination board.]

Chapter 6

Nandify

[General notes:

- Describe your approach, detail upon challenges, what were your struggles, how you came across them.
- If you did not manage to implement, also mention all your attempts and why your solution could not be implemented.
- Describe (in existent) additional features or smart computations that I might miss while assessing your code.
- All team work (2+ students) must be mentioned; e.g.: "In class Tree.cs from line 15-55 is code developed together with partner X (PCN:xxxxxx)".
- IMPORTANT: Please make sure to mention all lines of code you worked together with another fellow student, otherwise plagiarism will be detected in your code and you will be sent to the examination board.]

Chapter 7

Software design

[General notes:

- Give a general overview of your software design, your classes, interfaces, design patterns (UML diagram)
- Argue why your software design is representative to the assignment
- Describe why the choice of classes, interfaces, design patterns or mention why you decide not to choose certain classes, interfaces, design patterns and kept the code simple.
- Describe (in existent) additional feature or smart computations that I might miss while assessing your code.]

Chapter 8

GUI

[General notes:

- Give a general overview of your GUI (what does each components do).
- Argue why your GUI is representative to the assignment (easy to use, straightforward or require experienced user).
- Describe why the choice of components or mention why you decide not to choose certain components and kept the GUI simple.
- Describe (in existent) additional feature or smart computations that I might miss while assessing your code.]

Chapter 9

Testing

[General notes:

- Give a general overview of your testing (what does each test).
- Argue why your testing is representative to the project (enough - more than 10 tests per assignment - for each components)
- Describe the choice of testing or mention why you decide not to test certain components.
- Describe (in existent) additional feature or smart computations that I might miss while assessing your code.]

Chapter 10

Conclusions and future recommendations

[Conclude the report and mention future implementations, what could be improved.]

References

Swales, J. & Feak, C. (2012). *Academic writing for graduate students: Essential tasks and skills*. University of Michigan Press. Retrieved from https://books.google.nl/books?id=nL-nuAAACAAJ 1