CS6001 Formal Specification & Software Implementation

Coursework 1: Group Report

Finite State Automata

By: Duarte Pires Lopes **17028469**

Valeriu Bezer **18005806**

Gaurangkumar Gupta - **18005154**

Lauren Spruce - **18011848**

Table of Contents

Table of Figures

Tables

Introduction

Automata Theory is an exciting, theoretical branch of computer science that deals with designing abstract self-propelled computing devices. Finite automaton (used in text processing, compilers, hardware design) is the automaton with a finite number of states.

Finite Automaton can be classified into two types −

- Deterministic Finite Automaton (DFA)
- Non-deterministic Finite Automaton (NDFA / NFA)

Deterministic Finite Automaton (DFA)

In DFA, for each input symbol, one can determine the state to which the machine will move. Hence, it is called Deterministic Automaton. As it has a finite number of states, the machine is called Deterministic Finite Machine or Deterministic Finite Automaton.

An automaton can be represented by a 5-tuple $(Q, \sum, \delta, q_0, F)$, where −

- Q is a finite set of states.
- $\sum$ is a finite set of symbols, called the alphabet of the automaton.
- δ is the transition function.
- q_0 is the initial state from where any input is processed ($q_0 \in Q$).
- F is a set of final state/states of Q ($F \subseteq Q$).

Non-deterministic Finite Automaton (NFA)

In NDFA, for an input symbol, the machine can move to any combination of the states in the machine. In other words, the exact state to which the machine moves cannot be determined. Hence, it is called Non-deterministic Automaton. As it has a finite number of states, the machine is called Non-deterministic Finite Machine or Non-deterministic Finite Automaton.

An NFA can be represented by a 5-tuple $(Q, \sum, \delta, q_0, F)$ where −

- Q is a finite set of states.
- $\sum$ is a finite set of symbols, the alphabets.
- δ is the transition function where $\delta: Q \times \sum \rightarrow 2^Q$
  (Here the power set of Q (2^Q) has been taken because in case of NDFA, from a state, transition can occur to any combination of Q states)
- q_0 is the initial state from where any input is processed ($q_0 \in Q$).
- F is a set of final state/states of Q ($F \subseteq Q$).

DFA vs NFA

Table 1: DFA and NFA comparisons

DFA	NFA
The transition from a state is to a single next state for each input symbol. Hence it is called *deterministic*.	The transition from a state can be to multiple next states for each input symbol. Hence it is called *non-deterministic*.
Empty string transitions are not seen in DFA.	NDFA permits empty string transitions.
Backtracking is allowed in DFA	In NDFA, backtracking is not always possible.
Requires more space.	Requires less space.
A string is accepted by a DFA, if it transits to a final state.	A string is accepted by a NDFA, if at least one of all possible transitions ends in a final state.

Group Work

To complete our work, we created a WhatsApp group where all members of the team could communicate and discuss points of interest. Meetings were confirmed in this group, and discussion over what parts were left to be completed. We regularly communicated with one another through the duration of the coursework, constantly updating others on any new developments we had made to our group report.

Our work was stored on a google doc, this way everyone was able to edit and update the work in real time. Once all the basic information was added, we moved the final report onto a word document and did the final formatting and extra features there. This report was often posted in the Whatsapp group, so all team members could keep up to date, and give any feedback regarding it.

Meeting Logs

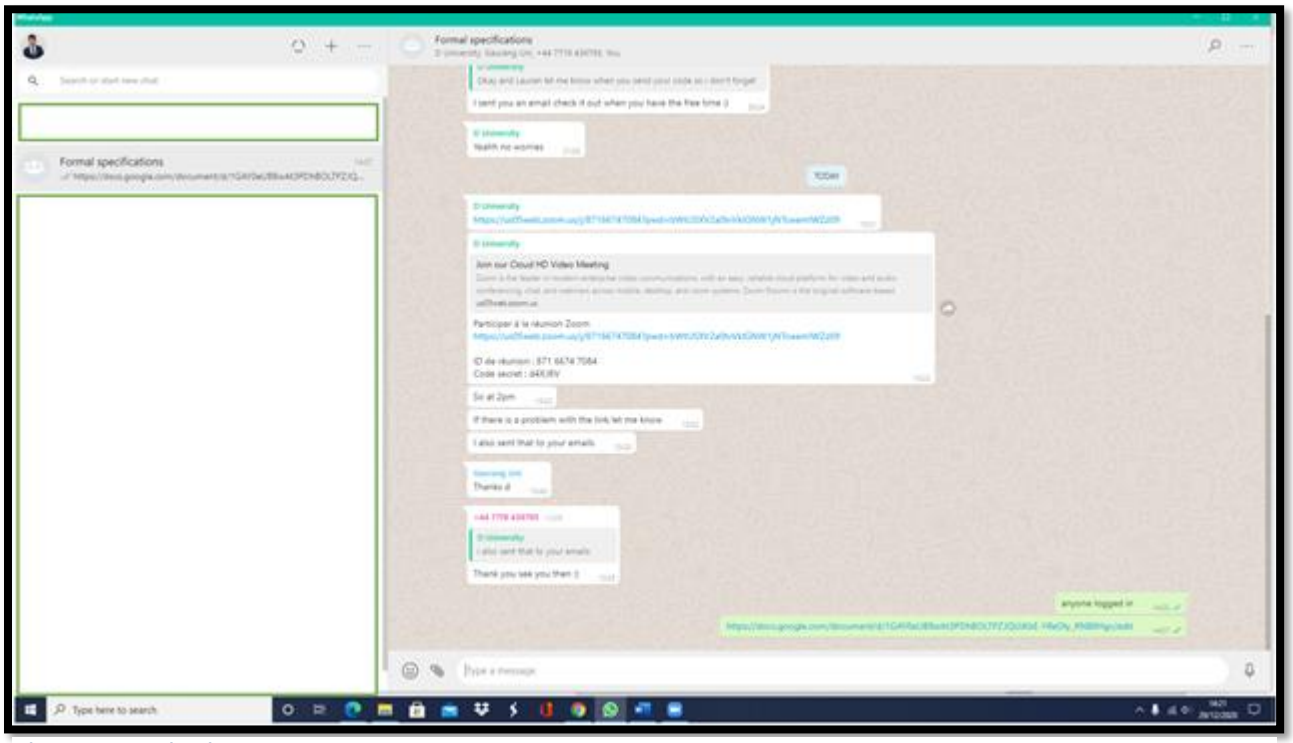

Figure 1: Meeting log 1

This screenshot shows the confirmed time to join the meeting to discuss the group assignment. All group members would agree to a specific time, if there was anyone unable to attend, we would reschedule the meeting to the next most suitable time.

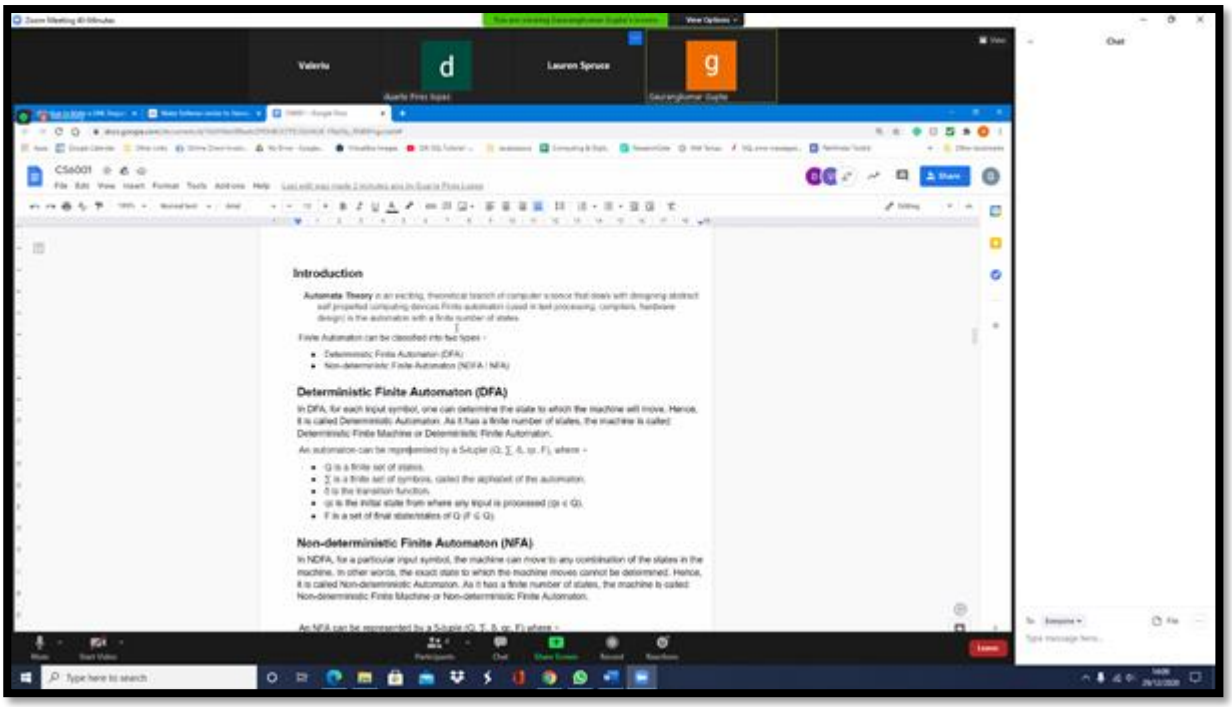

Figure 2: Meeting log 2

In this screenshot, it shows one of our regular meetings. Within this meeting, we discussed the group coursework and divided the tasks to work on. Everyone had a role to take part in, if anyone were unsure on their responsibilities this is where all team members would guide and support one another. We would also regularly keep up to date with the marking scheme, making sure we had covered every topic needed to polish our work.

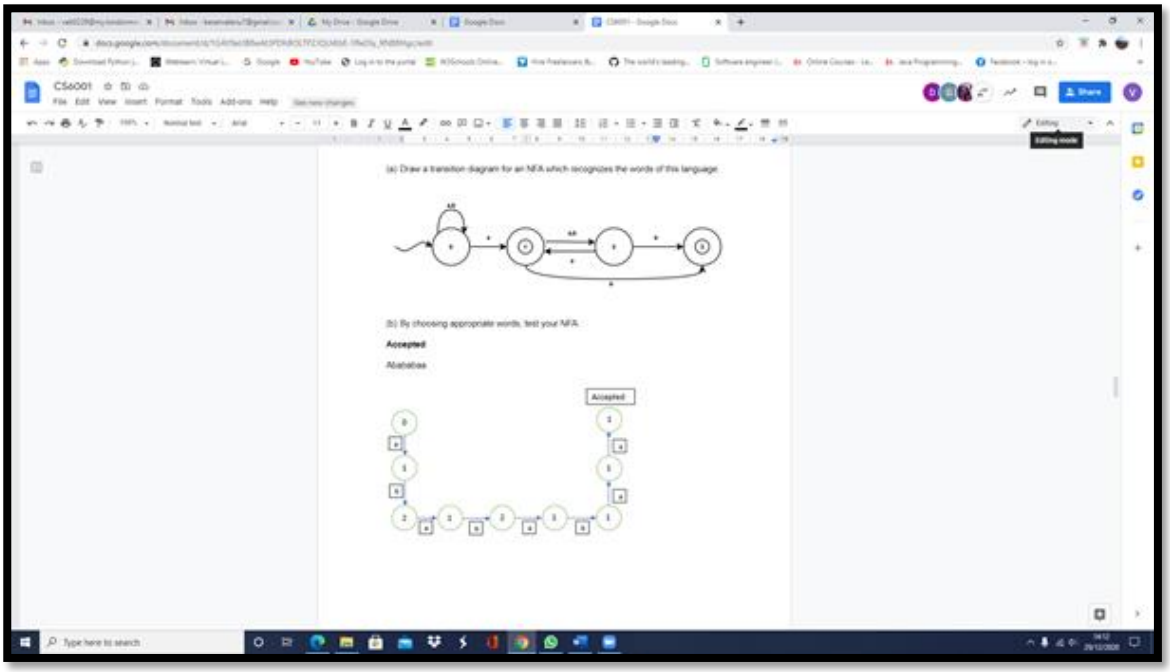

Figure 3: Meeting log 3

In this screenshot, it shows our google doc we created, where all our work was stored. All the work completed uploaded on the google document created where every member could see and contribute if any changes were required. During our conversations and discussions within Zoom and our WhatsApp group, this document was always referred to, to help us stay up to date with our progression.

Table 2 Meeting Log

Meeting	Agenda	Achievements	Discussion
1.	• Requirement analysis • NFA and DFA • Collaborative work Methodology • Tools • Report Structure	• Task Identification • Task Distribution • Whatsapp group • G - Drive Word Document • NFA Transition Diagram	• The point of discussion for this meeting was to identify the needs of the Group work and discuss the methodology of working together and achieving the results through using the right tools.
2	• NFA and DFA Language recognition • Word Testing • Transition Diagram • Conversion from NFA to DFA	• Tools Launched and used • Report Indexing and titles • Solution discussion on a,b,c and d part of the Group work section in coursework.	• This second meeting we came up with the results of the points discussed in previous meeting and had an intensive work together to come up with the solutions of a,b,c,d part for the Group coursework.
3	• DFA Minimization • Language testing • Task Distribution • Drafting Report	• Task e,f and g of the Group. • Word Testing and DFA Minimization • Transition Diagram • DFA – NFA conversion	• This meeting was to move to the next part of coursework requirement and revisit the achievement till task d. Both the DFA and NFA figures were checked with the transition diagram and the word acceptance of the language . Finalize the words recognition of accepted and rejected words . Discussion of drafting report and the task distribution related to that.
4	• Finalizing report • Requirements against achievements • Viva Preparation	• Revisiting all sections of the requirements • Testing and report discussion • Points to improve	• Final meeting to revisit all the points and requirements and results. Submission process , date, and improvements. Thanking and Congratulating each other on the group work.

Tasks

Transition Diagram for NFA

(a) Draw a transition diagram for an NFA which recognizes the words of this language.

The first step to find the NFA is to recognize the constraints in our language.

Our constraints are:

- The NFA can accept only a.
- The NFA only accept if a is part of the last three characters.
- The Length of the words can be less or bigger than three.

We start by building a start state with a loop a,b that will allow us to build any word over {a,b}. Next, we need to take care of the NFA last three characters and make sure a is part of it. The easiest way to create a list of accepted ending for our NFA

List of accepted ending for the NFA

a / aaa/ aab/ abb/ aba/ baa/ bba / bab

Once we have the list of accepted ending, the next step will be to create next states so we already the start state with a loop a,b now we will states that correspond to the language where the first of the last 3 characters start with a and where the first of the last 3 characters start b. Then we can reduce the NFA by asking ourselves if two states can be combined into 1. This is the result of the NFA with less states as possible.

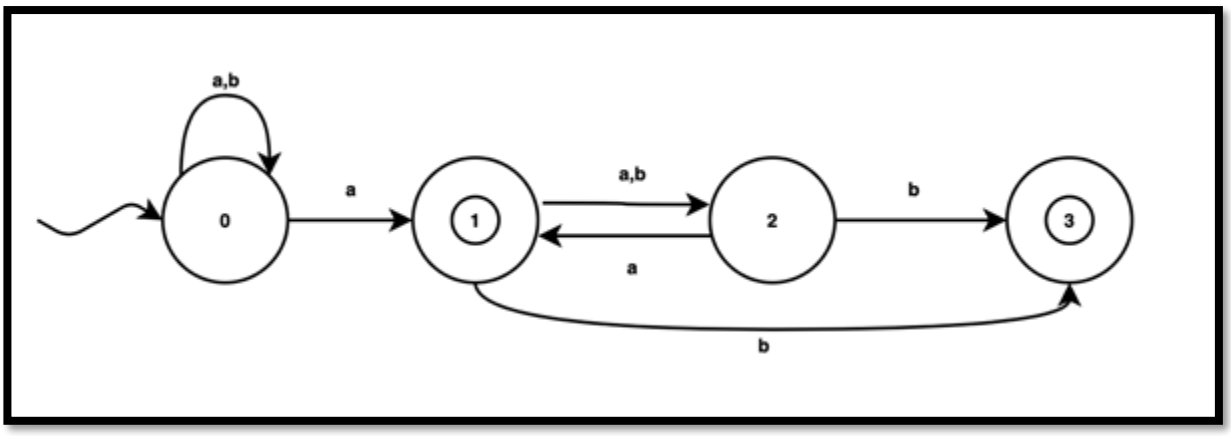

Figure 4: Transition Diagram for our NFA, which recognises language = {a,b}

Testing the NFA

Now, we are going to choose 6 appropriate words to test our NFA. We chose 3 words that would be accepted, and 3 words that would be rejected, created configuration sequences and diagrams to prove this.

These words are as listed:

1. abababaa
2. bbabb
3. aaaaaa
4. bbbbb
5. aabbb
6. babbb

Accepted words:

1. **abababaa**

Configuration Sequence and diagram:

0⊢bababaa 0⊢ababaa 0⊢babaa 0⊢abaa 0⊢baa 0⊢aa 0⊢a ⊢ 1 ∈ F = Accepted

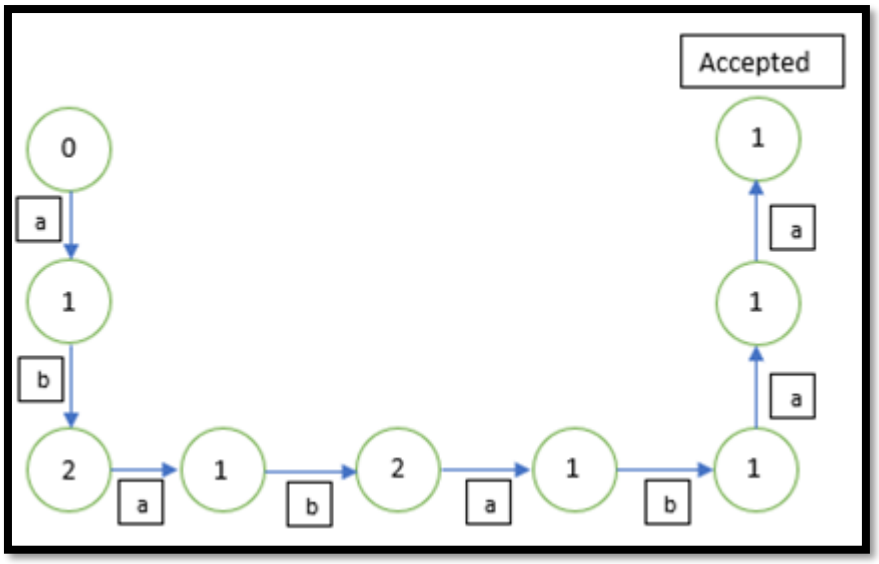

Figure 5: NFA: Configuration Sequence Diagram for abababaa

2. bbabb

Configuration Sequence and diagram:

0babb ⊢ 0abb ⊢ 1bb ⊢ 2b ⊢ 3 ∈ F = Accepted

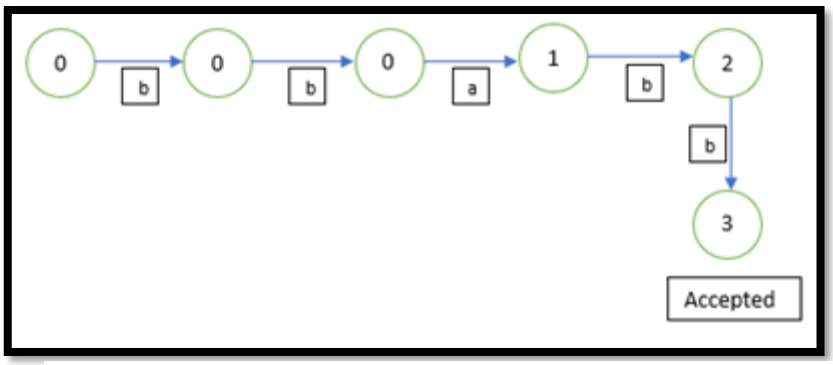

Figure 6: NFA: Configuration Sequence Diagram for Bbabb

3. aaaaaa

Configuration Sequence and diagram:

0aaaaaa ⊢1aaaaa ⊢ 2aaaa ⊢ 1aaa ⊢ 2aa ⊢ 1a ⊢ 1 ∈ F = Accepted

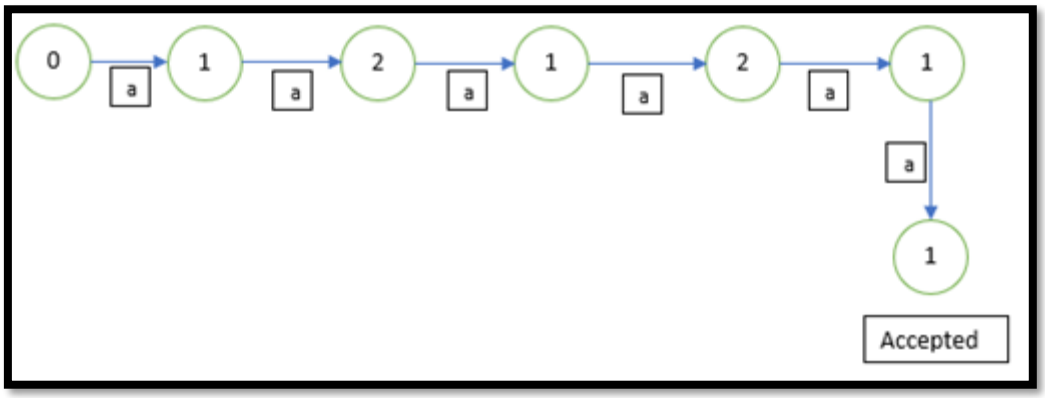

Figure 7: NFA: Configuration Sequence Diagram for aaaaaa

Refused Words:

1. **bbbbb**

Configuration Sequence and diagram:

0bbbbb ⊢ 0bbbb ⊢ 0bbb ⊢ 0bb ⊢ 0b ⊢ 0 ∉ = Rejected

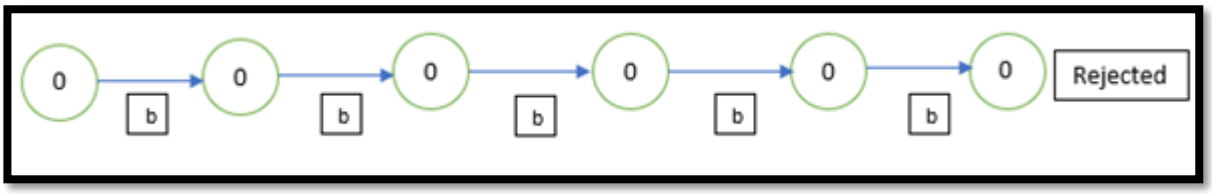

Figure 8: NFA: Configuration Sequence Diagram for bbbbb

2. **aabbb**

Configuration Sequence and diagram:

0aabbb ⊢ 0abbb ⊢ 0bbb ⊢ 0bb ⊢ 0b ⊢ 0 ∉ = Rejected

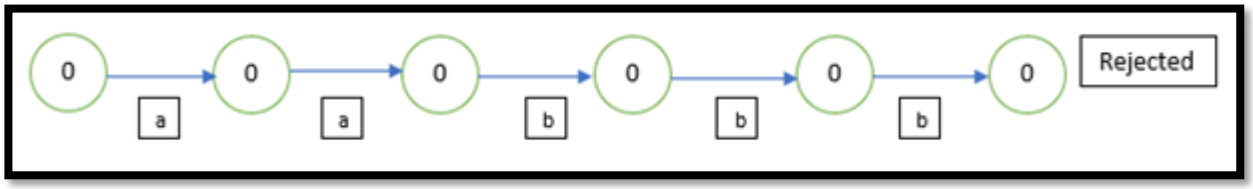

Figure 9: NFA: Configuration Sequence Diagram for aabbb

3. **babbb**

Configuration Sequence and diagram:

0babbb ⊢ 0abbb ⊢ 0bbb ⊢ 0bb ⊢ 0b ⊢ 0 ∉ = Rejected

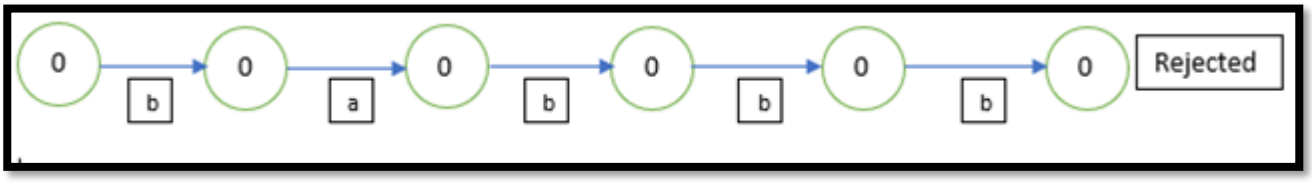

Figure 10: NFA: Configuration Sequence Diagram for babbb

Transition Relation of the NFA

In this section of our report, we created a transition table for our NFA. The states created within it, will be used to create the DFA.

To make the transition relation we want to look at our NFA and from each state will put the result of the transition into a set of state which will become a new state. For example, from state 0 a goes to 0 and 1 so we create the set {0,1}, we repeat the same process until there is now new state.

Table 3: Transition table for our NFA

		a	**b**
0’	{0}	{0,1}	{0}
1’	{0,1}	{0,1,2}	{0,2,3}
2’	{0,1,2}	{0,1,2}	{0,2,3}
3’	{0,2,3}	{0,1}	{0,3}
4’	{0,3}	{0,1}	{0}

Converting NFA into DFA

Using the transition table, we previously created, we created a DFA using all the NFA's states, this is now a completed DFA.

Now that our transition has been build, we can use it to convert our NFA to a DFA. We take new set of state and make them a state once we have our 4 set, we just need to map the transitions according to the table.

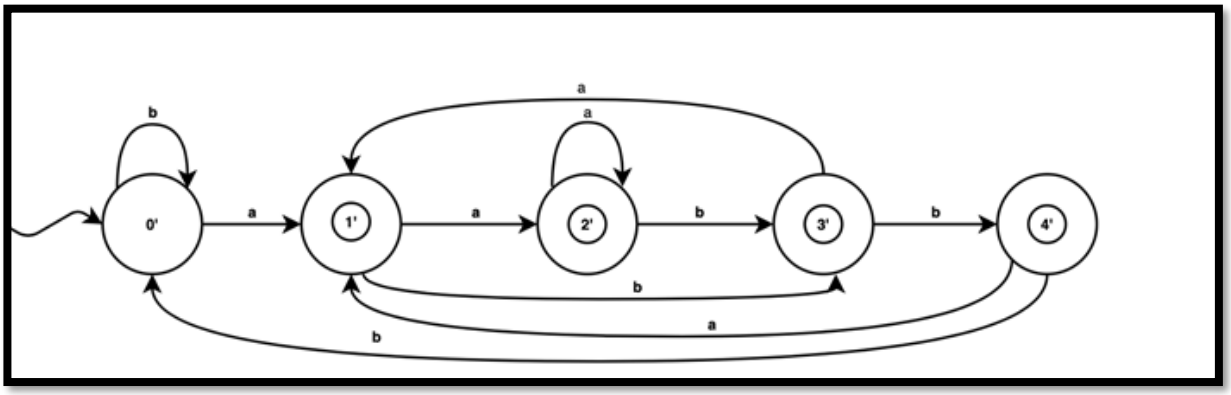

Figure 11: Converted DFA from original NFA

Transition Diagram for DFA

Now that we have created our DFA, we made a transition table, which used the new states from the DFA, we mapped all the new transitions created using the diagram and stored them within this table.

Table 4: Transition Diagram for our DFA

delta	a	b
0’	1’	0’
1’	2’	3’
2’	2’	3’
3	1’	4’
4	1’	0’

Minimising DFA

After we had completed both the transition table and diagram for the NFA, we now had to minimise it, to do this we needed to reduce the number of states within it. This was done by removing state 4'.

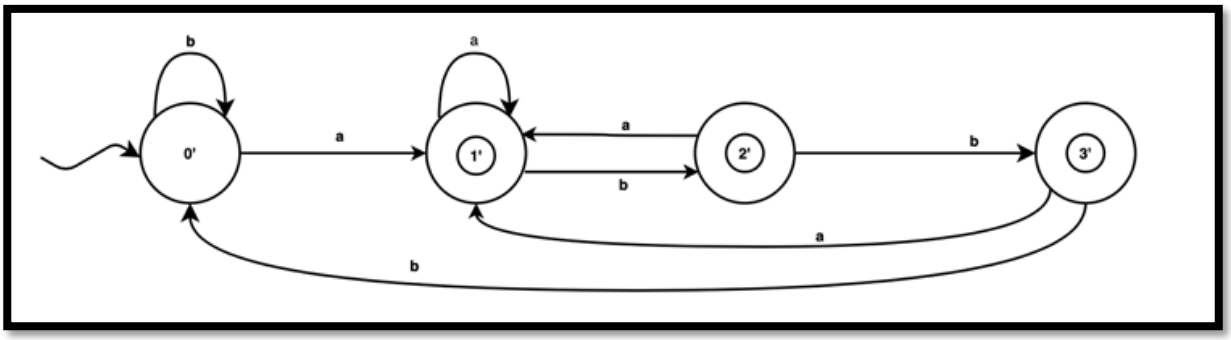

Figure 12: Minimised DFA from part e within our task

DFA Behaviour

Now that we had minimised the DFA, we needed to make sure that it would behave in the same manner as our original NFA, we did this by using the same words used previously to test the NFA, applying it to our new DFA.

Accepted words:

1. **abababaa**

Configuration Sequence and diagram:

$0abababaa \vdash 1bababaa \vdash 2ababaa \vdash 1babaa \vdash 2abaa \vdash 1baa \vdash 2aa \vdash 1a \vdash 1 \in F$ = Accepted

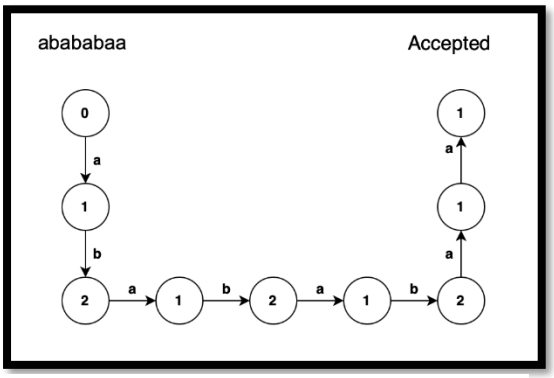

Figure 13: DFA: Configuration Sequence Diagram for abababaa

2. bbabb

Configuration Sequence and diagram:

$0bbabb \vdash 0babb \vdash 0abb \vdash 1bb \vdash 2b \vdash 3 \in F$ = Accepted

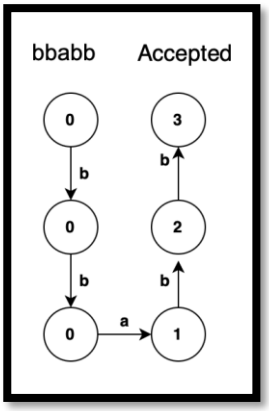

Figure 14: DFA: Configuration Sequence Diagram for bbabb

3. aaaaaa

Configuration Sequence and diagram:

$0aaaaaa \vdash 1aaaa \vdash 1aaa \vdash 1aa \vdash 1a \vdash 1 \in F$ = Accepted

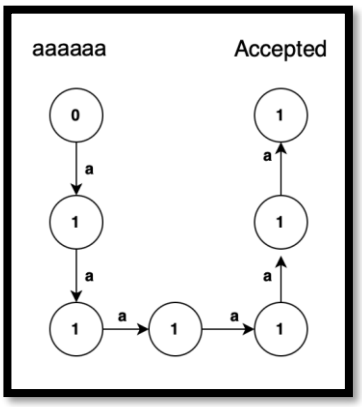

Figure 15: DFA: Configuration Sequence Diagram for aaaaaa

Rejected

1. **bbbbb**

Configuration Sequence and diagram:

0bbbbb ⊢ 0bbbb ⊢ 0bbb ⊢ 0bb ⊢ 0b ⊢ 0 ∉ F = Rejected

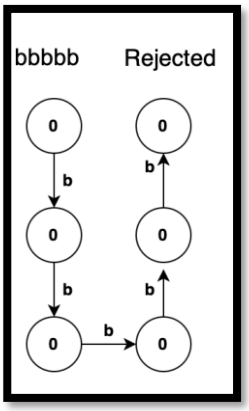

Figure 16: DFA: Configuration Sequence Diagram for bbbbb

2. **aabbb**

Configuration Sequence and diagram:

0aabbb ⊢ 1abbb ⊢ 1bbb ⊢ 2bb ⊢ 3b ⊢ 0 ∉ F = Rejected

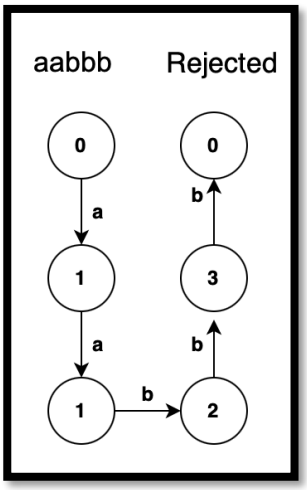

Figure 17: DFA: Configuration Sequence Diagram for aabbb

3. babbb

Configuration Sequence and diagram:

0babbb ⊢ 0abbb ⊢ 1bbb ⊢ 2bb ⊢ 3b ⊢ 0 ∉ F = Rejected

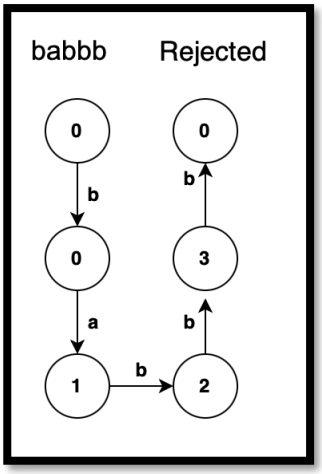

Figure 18: DFA: Configuration Sequence Diagram for babbb

Conclusion

The main goal of this report was to create a NFA for a specify language. Once the NFA was created, more operations could be performed. This report started by showing the words that the NFA could accept and then the words would be rejected. To follow, a transition relation table is shown where the different transitions can be observed and from this data, we produced a DFA with five states. With the DFA establish the next step was to produce a transition diagram for the DFA. Then, the process of minimisation was applied to the DFA by removing the state number two. To finish the last task was to use same words chosen at the beginning for the NFA and test them on DFA.

This report extends the understanding of how to take a language with a certain constraint and be able to construct the NFA that corresponds to it. It also shows that having one constraint in this case having {a} be part of the last three can lead to many other constraints which was interesting.

What went well was that all team members engaged consistently throughout the whole project, the WhatsApp group improved our group activity a lot as everyone had a way of engaging with the work. The use of google docs also improved the efficiency of our workflow, as everything stayed up to date in real time. Zoom meetings helped us a lot as we were able to share content with one another and discussion important issues that needed resolving in our coursework.

Sometimes the tasks could be confusing, like minimising the DFA. To overcome this, we asked for guidance from our lecturer and tutor, we also revised lecture notes for each specific topic needed to complete our individual tasks. Things that we could improve would be to have spent more time on annotating diagrams throughout the report, perhaps offering more research into DFAs and NFAs a whole.

Overall, this coursework has been very engaging, it has helped reinforce many of the topics and learning objectives we have covered, providing extra stability and support in future coursework’s and exams.