1 DFA Minimisation

In this assignment we are asked to minimise **??** in accordance to the minimisation steps presented in the lecture. Unfortunately I was unable to attend the lecture, but from the lecture slides, it appears that the method specified in the book was followed.
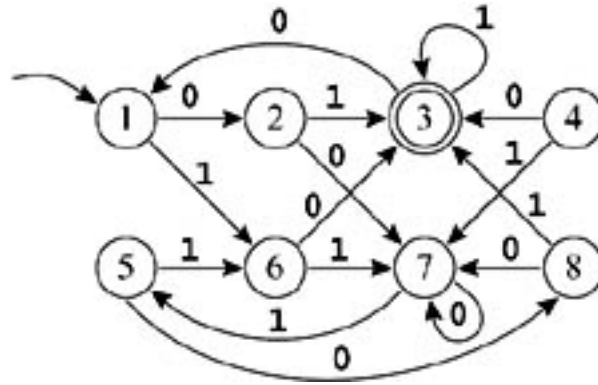

Figure 1: DFA to minimise

In order to minimise this DFA we need to identify the different groups, check them and mark them accordingly. The first thing we do is create a subset of our nodes without accepting nodes S
$F = 1, 2, 4, 5, 6, 7, 8$ and a set of our accepting nodes $F = 3$.
Being that there is only one node in our set $G1 = F$ we know that this group is consistent and can mark it without checking it. We name our set $G2 = S$
F. Now we wish to check this group for consistency. Upon an inspection of the DFA we can conclude that there are no dead groups.

G2	0	1
1	G2	G2
2	G2	G1
4	G1	G2
5	G2	G2
6	G1	G2
7	G2	G2
8	G2	G1

Table 1: Checking G2 for consistency

This group is obviously not consistent, so we split it into groups of that

1

are maximally consistent. By maximally consistent groups, we mean that we split $G2$ into, in this case two, groups that are seemingly consistent. As we do this all marked groups are unmarked and checked again. $G1$ is still a singleton group and thus still consistent and we can mark it. $G2$ is split into the groups $G3 = 1, 5, 7$, $G4 = 2, 8$, and $G5 = 4, 6$. We then check if $G3$ is consistent.

G3	0	1
1	G4	G5
5	G4	G5
7	G3	G3

Table 2: Checking G3 for consistency

Obviously $G3$ isn't consistent, so it is further split into two group $G6 = 1, 5$ and $G7 = 7$.

G6	0	1
1	G4	G7
5	G4	G7

Table 3: Checking G6 for consistency

$G6$ is consistent, so we add it to our set of consistent groups $S = G1, G6$.
$G4$ is consistent, so we add it to the set of consistent groups $S = G1, G6, G4$.

G4	0	1
2	G7	G1
8	G7	G1

Table 4: Checking G4 for consistency

$G5$ is consistent (see table **??**), so we add it to the set of consistent groups $S = G1, G6, G4, G5$.
Being that $G7$ is a singleton group, this is consistent so we can add it to our set of consistent groups $S = G1, G6, G4, G5, G7$ and thus we are done as there are no more unmarked groups.

G5	0	1
4	G1	G7
6	G1	G7

Table 5: Checking G5 for consistency
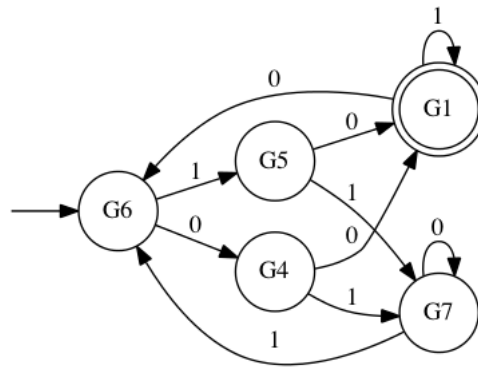
The minimised DFA will then look as follows:

Figure 2: "Minimised DFA"

2 Backtracking Automaton

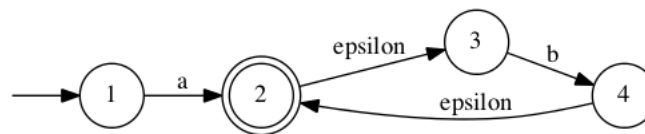## 2.1 Give an NFA for each expression $\alpha1, \alpha2, \alpha3$ and combined NFA
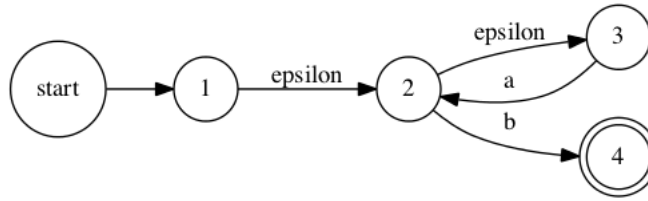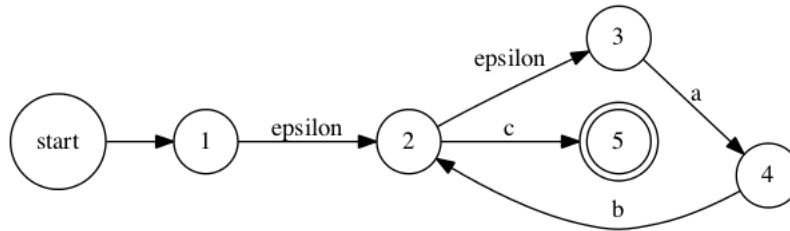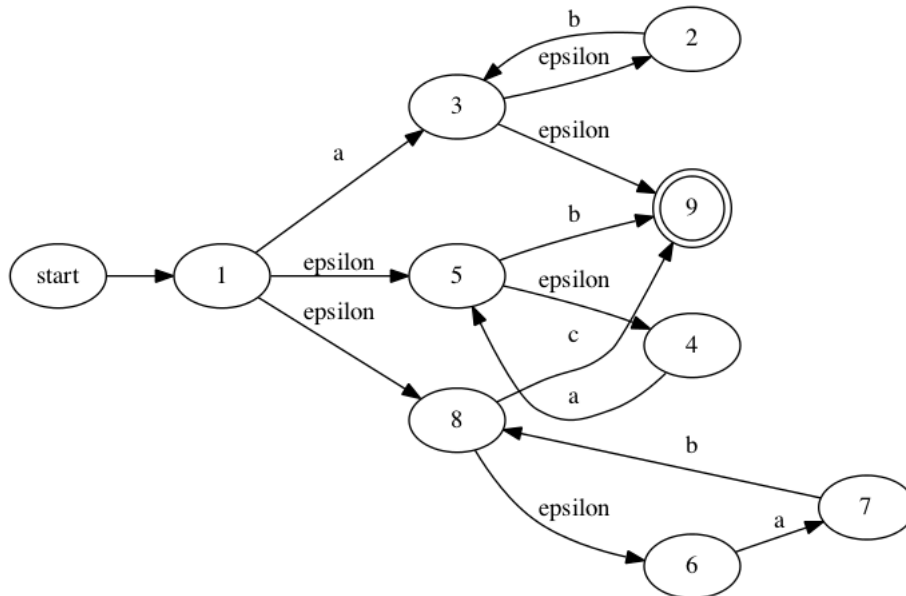

Figure 3: "a1 NFA"

Figure 4: "a2 NFA"

caption"a3 NFA"

Figure 5: "Combined NFA"

2.2 Convert the combined NFA into a DFA

We want to find the inital epsilon transitions from the starting point.

$$\epsilon - closure(\{1\}) = \{1, 5, 8, 4, 6\}$$

4

Now we would like to find the states of the DFA, which is done by subset construction:

$$move(s_0, a) = \{2, 3, 9, 4, 5, 7\} = s_1$$

$$move(s_0, b) = \{9\} = s_2$$

$$move(s_0, c) = \{9\} = s_3$$

$$move(s_1, b) = \{3, 2, 9, 8, 6\} = s_4$$

$$move(s_4, c) = \{9\} = s_5$$

$$move(s_4, a) = \{7\} = s_6$$

$$move(s_6, b) = \{8, 6\} = s_7$$
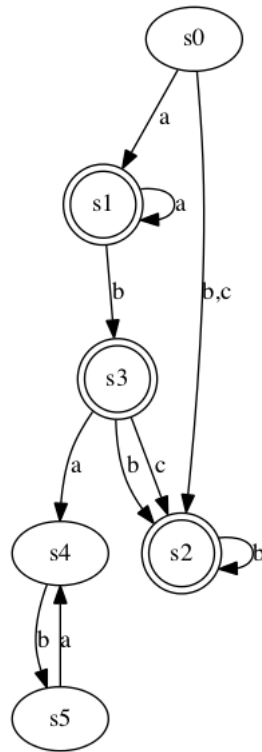

Figure 6: "Result of NFA to DFA result"

2.3 Describe the transitions and backtracking

First we move as follows: $s0 - s1 - s3 - s4 - s5 - s4$. However only the initial two transitions are saved, and the remaining is sent back to the input stream, thus we have currently parsed "ab".

Then we repeat the movement, but stop after the first time we reach $s4$ giving us: "ab". We then move $s0 - s1$ after which we cannot omve futher and restart from the initial state, we have obtained "a".

We now move $s0 - s2$, to obtain a "c". We do this twice, we now have "cc". Finally we move $s0 - s1$ to obtain the final "a", thus we have now parsed the entire input "ababacca".

3 Mosmllex scanner construction

4 More on regular languages and tokenization

4.1 Using the alphabet of decimal digits, give regular expression describing given languages

- Write a regular expression describing numbers divisible by 5.
  $[0-9]^*[0|5]^+$

- Write a regular expression describing numbers where the number 5 occurs exactly three times.
  $[0-4] * [6-9] * [0-4] * 5[0-4] * [6-9] * [0-4] * 5[0-4] * [6-9] * [0-4] * 5[0-4] * [6-9] * [0-4]*$

4.2 Are the following languages over the alphabet of decimal digits regular? Give short convincing reasons for your answers

- Numbers which contain digit '1' exactly as many times as digit '2'.
  It isn't a regular language, as isn't bounded, which a regular language must be.

- Numbers N ¡ 1.000.000 which contain digit '1' exactly as often as digit '2'.
  Yes. It can be described as we are working with a finite sequence, and can thus implement rules, around which we build our regular expression. I have not figured out the regular expression though.