Opgave 3.2 - Programmer som data

Albert Ross Johannessen
Casper Storm Frøding
Mathilde Julie Gonzalez-Knudsen

September 12, 2024

1

1.1 Regexp

We have assumed that the empty string not is allowed, due to the specification that it must consist of the letters A and B.
With these assumptions in mind, we have found the regex.

$$\big(a?(ba?)+\big)|a \tag{1}$$

1.2 NFA

From the regexp seen in equation 1, we have constructed a NFA that is Figure 1 longer down. We have used the method from the book, to convert a regexp to a NFA, and then simplified the edges. This being that we removed redundant edges, and also combined some Epsilon edges with some letters, while still holding the logic of the NFA.

1.3 DFA

From the NFA we have constructed the corresponding DFA to this. This is the figure 3.
To do this, we used the method shown in the lecture, to convert an NFA to a DFA. The skema for this conversion is the scheme in figure 2.

1.4 Simplified DFA

In Figure 4, the nodes from the DFA, have been seperated into final states, and non final states, this is Figure 4 G1 is already single, and can therefore not be simplified. Therefore we only make the scheme for G2, which is in Figure 5. As can be seen in the figure, this is not consistent, so we have to split it into two new groups, which is done in Figure 6.
Now all groups only have one node, and therefore cannot be simplified more. Hence we have proved that this cannot be simplified more then it was already.

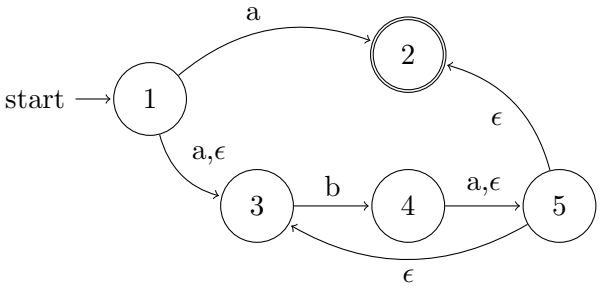

Figure 1: The labeled NFA representing $\big(a?(ba?)+\big)|a$

	a	b	NFA state
S_1	$\{2\}^{S_2}$	$\{4\}^{S_3}$	$\{1,3\}$
S_2	$\{\}$	$\{4\}^{S_3}$	$\{1,2,3\}$
S_3	$\{2\}^{S_2}$	$\{\}^{S_3}$	$\{1,2,3,4\}$

Figure 2

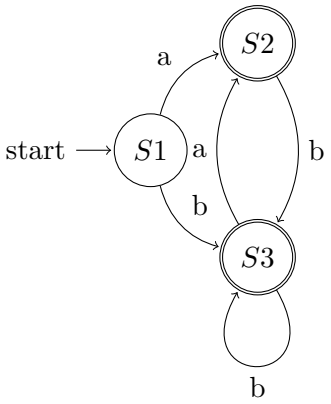

Figure 3: The labeled DFA representing $\big(a?(ba?)+\big)|a$

G_1	$\{S_1\}$
G_2	$\{S_2, S_3\}$

Figure 4

G_2	a	b
S_2	$-$	$\{G_2\}$
S_3	$\{G_2\}$	$\{G_2\}$

Figure 5

This is not consistent, so we have to split it into two new groups:

G_1	$\{S_1\}$
G_3	$\{S_2\}$
G_4	$\{S_3\}$

Figure 6