

Compilers: Assignment #4

Due on Sunday, January 10, 2016

Resubmit: Task 2 and as much as possible Task3

Mirza Hasanbasic

Indhold

Task 1	2
Task 2	3
Task 3	4

Task 1

Først defineres $s'_0 = \hat{e}(s_0) = \{1, 2, 4\}$

Det vil sige, at når vi kigger på move fra start gennem enten a eller b, kan vi benytte at vi er i punkt 1, 2 eller 4

$$\begin{aligned} \text{move}(\{s'_0, a\}) &= \{3, 4\} = \{3, 4, 6\} = s'_1 \\ \text{move}(\{s'_0, b\}) &= \{5\} = \{5, 4, 6\} = s'_2 \end{aligned}$$

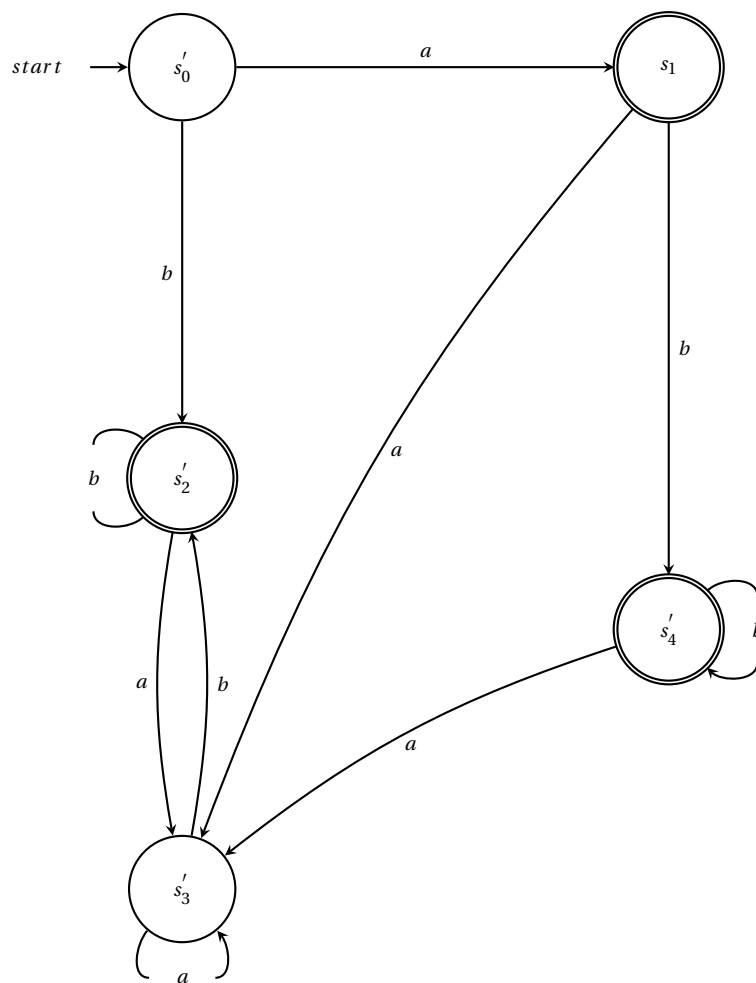
Da vi nu har fået to punkter og tjekket s_0 , kan vi gå videre til s_1 og s_2 .

$$\begin{aligned} \text{move}(\{s'_1, a\}) &= \{4\} = s'_3 \\ \text{move}(\{s'_1, b\}) &= \{3, 5\} = \{3, 5, 4, 6\} = s'_4 \\ \text{move}(\{s'_2, a\}) &= \{4\} = s'_3 \\ \text{move}(\{s'_2, b\}) &= \{5\} = \{5, 4, 6\} = s'_2 \end{aligned}$$

Vi tjekker de sidste 2, nemlig s_3 og s_4

$$\begin{aligned} \text{move}(\{s'_3, a\}) &= \{4\} = s'_3 \\ \text{move}(\{s'_3, b\}) &= \{5\} = \{5, 4, 6\} = s'_2 \\ \text{move}(\{s'_4, a\}) &= \{4\} = s'_3 \\ \text{move}(\{s'_4, b\}) &= \{3, 5\} = \{3, 5, 4, 6\} = s'_4 \end{aligned}$$

Da $\{s_1, s_2, s_4\}$ inderholder ending state 6, vil det derfor være at $F' = \{s_1, s_2, s_4\}$



Figur 1: DFA repræsentationen

Task 2

Først oprettes en tabel, for at kunne se hvilke states der er forbundet. Hvor der står 3^* , betyder blot at det er ending state.

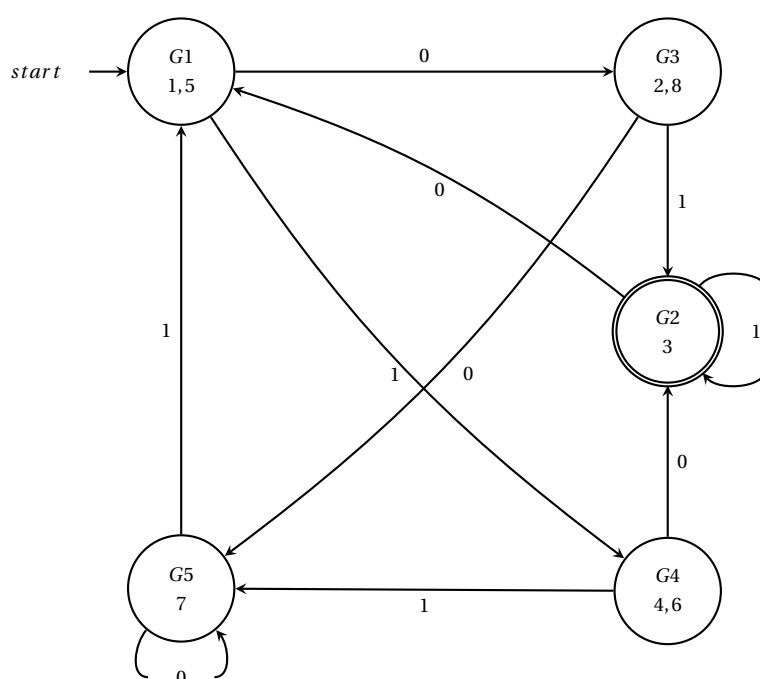
	0	1
1	2	6
2	7	3^*
3^*	1	3^*
4	3^*	7
5	8	6
6	3^*	7
7	7	5
8	7	3^*

Nu sammenlignes der for hvilke indgange der kommer i andre grupper. Mange sammenligner (q_i, q_j) mod hinanden og ser om de er i samme eller anden gruppe iterativt. Vi starter med 0-equal, hvor ending state vil være G2 og alle andre vil være G1.

	G1	G2	G3	G4	G5
0-equal	1,2,4,5,6,7,8	3			
1-equal	1,5,7	3	2,8	4,6	
2-equal	1,5	3	2,8	4,6	7
3-equal	1,5	3	2,8	4,6	7

Altså er DFA minimeret og ser ud som følger

FIX:



Figur 2: Hvor start noden bliver angivet og slutnoden er dobbelcirkel

Task 3

FIX:

a)

i)

1 [0-4 6-9][0-9]*(0|5)

ii)

```
1 ^[\D5]*5[\D5]*5[\D5]*5[\D5]*$
2
3 // ^           from start
4 // [\D5]*      not a non-digit or 5
5 // 5           the first '5' digit
6 // $           till the end
```

b)**i)**

Det er ikke muligt at have en regular expression der matcher præcis ethvert nummer med præcis antal af 1 eller 2 siden der findes uendelig mange numre og vi ikke har nogen variable til at blive ved med at tælle antallet af karaktere

ii)

Antag at vi kun ønsker at matche ikke-negative numre, så er der kun et endelig mængde af numre, mellem 0 og 1.000.000 hvor forekomsten af 1 og 2 er det samme. Derfor kan man identificere alle disse numre og lave en match præcis med disse numre.