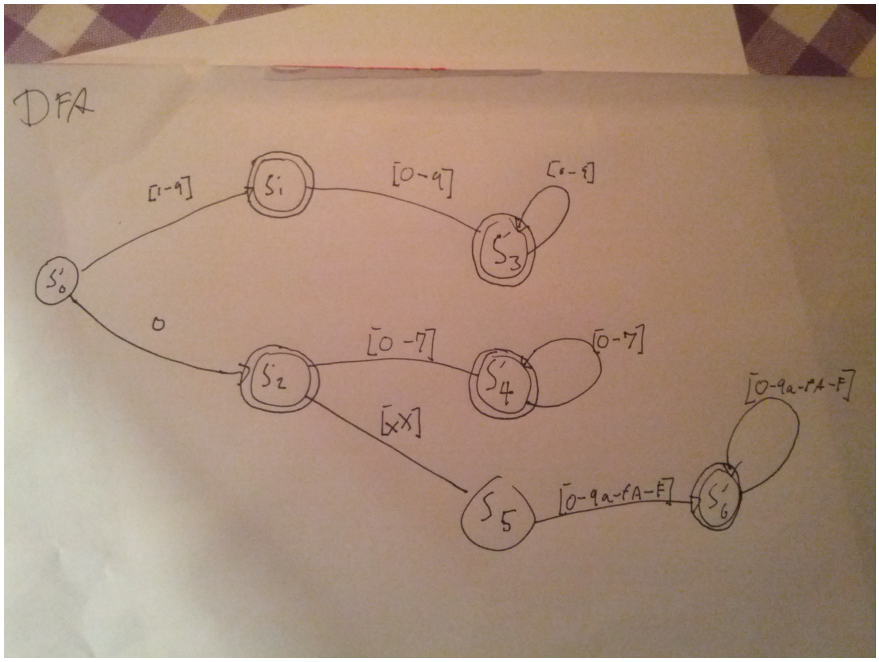
W1 - Oversættere

2. december 2012

1 Automata Recognising Number Literals

1.1 Draw a DFA

Fra den oprindelige aflevering, havde jeg faktisk en skitse liggende som jeg tegnede tidligt i processen. Den er identisk med DFA'en som er udledt i opgave 1.(b-c).



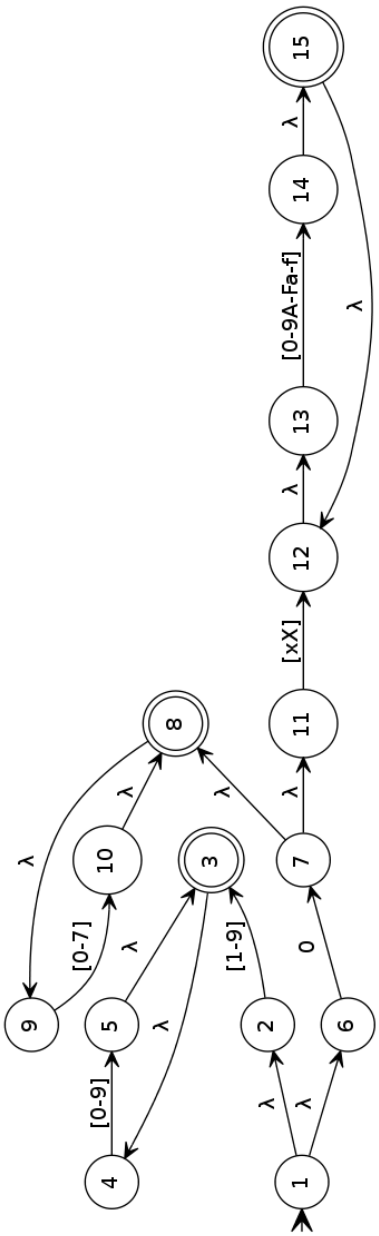
Figur 1:

1.2 Regex to NFA

Nedenfor ses det regulerer udtryk som konverteres til en NFA.

$$(([1-9][0-9]*|0)|0([0-7]*|[xX][0-9a-fA-F]+))$$

NFA'en kan ses i Figur 1.



Figur 2: NFA konstrueret fra det regulerer udtryk ovenfor.

1.3 Konvetering til DFA

$$\hat{\varepsilon}\{1\} = \{1, 2, 6\}$$

$S = \{s_0'\}$

$$\begin{aligned} move(s'_0, [1-9]) &= \hat{\varepsilon}\left(\left\{t | s \in \{1,2,6\} \text{ and } s^{[1-9]}t \in T\right\}\right) \\ &= \hat{\varepsilon}(\{3\}) \\ &= \{3,4\} \\ &= s'_1 \end{aligned}$$

$$\begin{aligned} move(s'_0, 0) &= \hat{\varepsilon}\left(\{t | s \in \{1,2,6\} \text{ and } s^{0}t \in T\}\right) \\ &= \hat{\varepsilon}(\{7\}) \\ &= \{7,8,9,11\} \\ &= s'_2 \end{aligned}$$

$S = \{\overset{\checkmark}{s'_0}, s'_1, s'_2\}$

$$\begin{aligned} move(s'_1, [0-9]) &= \hat{\varepsilon}\left(\left\{t | s \in \{3,4\} \text{ and } s^{[0-9]}t \in T\right\}\right) \\ &= \hat{\varepsilon}(\{5\}) \\ &= \{3,4,5\} \\ &= s'_3 \end{aligned}$$

$S = \{\overset{\checkmark}{s'_0}, \overset{\checkmark}{s'_1}, s'_2, s'_3\}$

$$\begin{aligned} move(s'_2, [0-7]) &= \hat{\varepsilon}\left(\left\{t | s \in \{7,8,9,11\} \text{ and } s^{[0-7]}t \in T\right\}\right) \\ &= \hat{\varepsilon}(\{10\}) \\ &= \{8,9,10\} \\ &= s'_4 \end{aligned}$$

$$\begin{aligned} move(s'_2, [\text{Xx}]) &= \hat{\varepsilon}\left(\left\{t | s \in \{7,8,9,11\} \text{ and } s^{[\text{Xx}]}t \in T\right\}\right) \\ &= \hat{\varepsilon}(\{12\}) \\ &= \{12,13\} \\ &= s'_5 \end{aligned}$$

$S = \{\overset{\checkmark}{s'_0}, \overset{\checkmark}{s'_1}, \overset{\checkmark}{s'_2}, s'_3, s'_4, s'_5\}$

$$\begin{aligned} move(s'_3, [0-9]) &= \hat{\varepsilon}\left(\left\{t | s \in \{3,4,5\} \text{ and } s^{[0-9]}t \in T\right\}\right) \\ &= \hat{\varepsilon}(\{5\}) \\ &= \{3,4,5\} \\ &= s'_3 \end{aligned}$$

$S = \{\overset{\checkmark}{s'_0}, \overset{\checkmark}{s'_1}, \overset{\checkmark}{s'_2}, \overset{\checkmark}{s'_3}, s'_4, s'_5\}$

$$
\begin{aligned}
move(s'_4, [0-7]) &= \hat{\varepsilon}\left(\left\{t | s \in \{8,9,10\} \text{ and } s^{[0-7]}t \in T\right\}\right) \\
&= \hat{\varepsilon}(\{10\}) \\
&= \{8,9,10\} \\
&= s'_4
\end{aligned}
$$

$S = \{\overset{\checkmark}{s'_0}, \overset{\checkmark}{s'_1}, \overset{\checkmark}{s'_2}, \overset{\checkmark}{s'_3}, \overset{\checkmark}{s'_4}, s'_5\}$

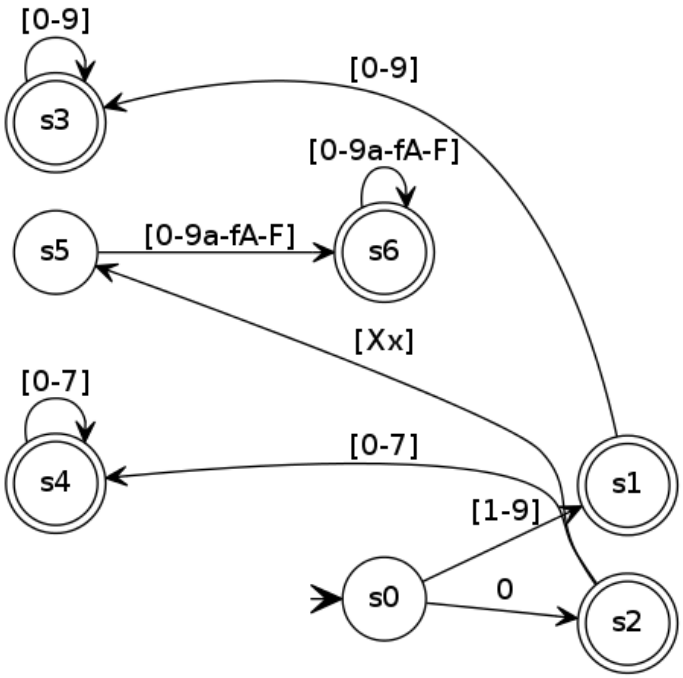
$$
\begin{aligned}
move(s'_5, [\text{0-9a-fA-F}]) &= \hat{\varepsilon}\left(\left\{t | s \in \{12,13\} \text{ and } s^{[\text{0-9a-fA-F}]}t \in T\right\}\right) \\
&= \hat{\varepsilon}(\{14\}) \\
&= \{12,13,14,15\} \\
&= s'_6
\end{aligned}
$$

$S = \{\overset{\checkmark}{s'_0}, \overset{\checkmark}{s'_1}, \overset{\checkmark}{s'_2}, \overset{\checkmark}{s'_3}, \overset{\checkmark}{s'_4}, \overset{\checkmark}{s'_5}, s'_6\}$

$$
\begin{aligned}
move(s'_6, [\text{0-9a-fA-F}]) &= \hat{\varepsilon}\left(\left\{t | s \in \{12,13,14,15\} \text{ and } s^{[\text{0-9a-fA-F}]}t \in T\right\}\right) \\
&= \hat{\varepsilon}(\{14\}) \\
&= \{12,13,14,15\} \\
&= s'_6
\end{aligned}
$$

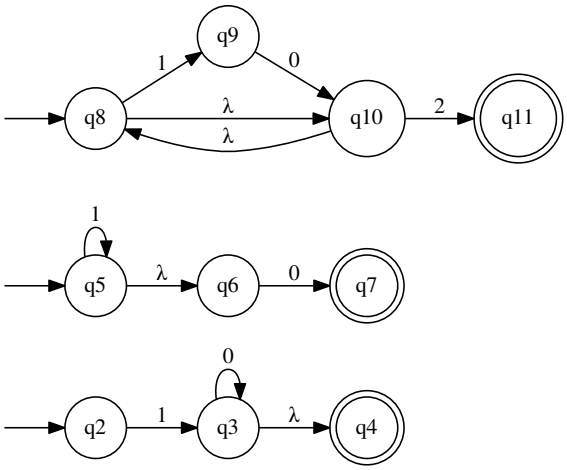
$S = \{\overset{\checkmark}{s'_0}, \overset{\checkmark}{s'_1}, \overset{\checkmark}{s'_2}, \overset{\checkmark}{s'_3}, \overset{\checkmark}{s'_4}, \overset{\checkmark}{s'_5}, \overset{\checkmark}{s'_6}\}$

Her er jeg færdig og har lavet min DFA med seks states.

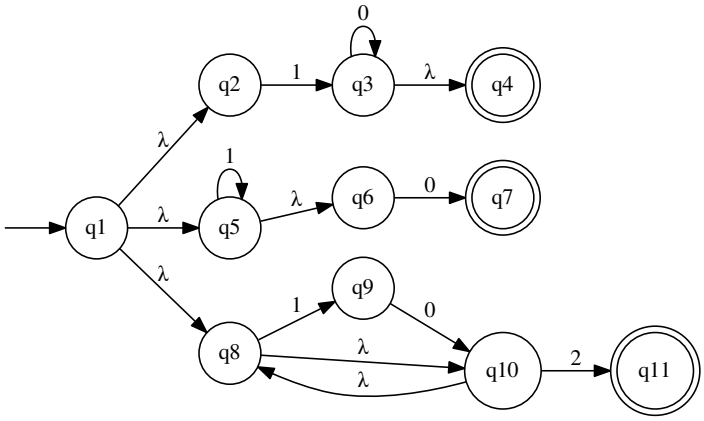


Figur 3: DFA, konveterert fra NFA i opgave 1.2

2 Backtracking Automaton



Figur 4: NFA's (before combine) for the three regular expressions given by the assignment.



Figur 5: The NFA's from Figure 5 combined to a new single NFA

Converting NFA in Figure 5 to DFA

$$\hat{\lambda}\{q1\} = \{q1, q2, q5, q6, q8, q10\} = s'_0$$

$$\begin{aligned}
move(s'_0, 0) &= \hat{\lambda}(\{q7\}) = \{q7\} = s'_1 \\
move(s'_0, 1) &= \hat{\lambda}(\{q3, q5, q9\}) = \{q3, q4, q5, q6, q9\} = s'_2 \\
move(s'_0, 2) &= \hat{\lambda}(\{q11\}) = \{q11\} = s'_3
\end{aligned}$$

$$\begin{aligned}
move(s'_2, 0) &= \hat{\lambda}(\{q3, q7, q10\}) = \{q3, g4, q7, q8, q10\} = s'_4 \\
move(s'_2, 1) &= \hat{\lambda}(\{q5\}) = \{q5, g6\} = s'_5
\end{aligned}$$

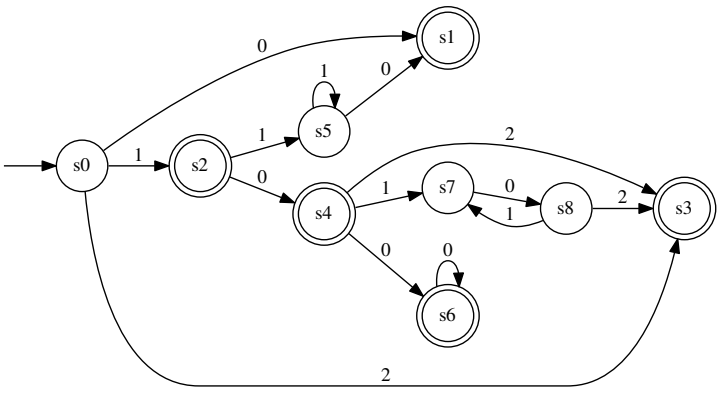
$$\begin{aligned}
move(s'_4, 0) &= \hat{\lambda}(\{q3\}) = \{q3, g4\} = s'_6 \\
move(s'_4, 1) &= \hat{\lambda}(\{q9\}) = \{q9\} = s'_7 \\
move(s'_4, 2) &= \hat{\lambda}(\{q11\}) = \{q11\} = s'_3
\end{aligned}$$

$$\begin{aligned}
move(s'_5, 0) &= \hat{\lambda}(\{q7\}) = \{q7\} = s'_1 \\
move(s'_5, 1) &= \hat{\lambda}(\{q5\}) = \{q5, q6\} = s'_5
\end{aligned}$$

$$move(s'_6, 0) = \hat{\lambda}(\{q3\}) = \{q3, q4\} = s'_6$$

$$move(s'_7, 0) = \hat{\lambda}(\{q10\}) = \{q8, q10\} = s'_8$$

$$\begin{aligned}
move(s'_8, 1) &= \hat{\lambda}(\{q9\}) = \{q9\} = s'_7 \\
move(s'_8, 2) &= \hat{\lambda}(\{q11\}) = \{q11\} = s'_3
\end{aligned}$$



Figur 6:

2.1 Transitions and backtracking

$$s_0\;1\;s_2\;0\;\underbrace{s_4\;1\;s_7\;0\;s_8\;1\;s_7}_{s_4 \leftarrow s_7} \quad | \quad 2 \quad 2 \quad 1$$

$$\underline{10} \quad s_0\;1\;s_2\;0\;\underbrace{s_4\;1\;s_7}_{s_4 \leftarrow s_7} \quad | \quad 2 \quad 2 \quad 1$$

$$\underline{10} \quad \underline{10} \quad s_0\;1\;s_2 \quad | \quad 2 \quad 2 \quad 1$$

$$\underline{10} \quad \underline{10} \quad \underline{1} \quad s_0\;2\;s_3 \quad | \quad 2 \quad 1$$

$$\underline{10} \quad \underline{10} \quad \underline{1} \quad \underline{2} \quad s_0\;2\;s_3 \quad | \quad 1$$

$$\underline{10} \quad \underline{10} \quad \underline{1} \quad \underline{2} \quad \underline{2} \quad s_0\;1\;s_2 \quad |$$

$$\underline{10} \quad \underline{10} \quad \underline{1} \quad \underline{2} \quad \underline{2} \quad \underline{1}$$

3 Lexer i SML

A)

Regular expresson for the English time format presented in the assignment.

```
((quarter|half) past ([1-9]|10|11|12)|([1-9]|10|11|12) o'clock
    |([0-9]|[1-5][0-9]) to ([1-9]|10|11|12))
```

B)

The solution can be found in the files 3b.lex, 3b.sml and 3b_test.sml.

4 More on Regular Languages and Tokenisation

A)

1. Ja, da vi leder efter tal der slutter på 0 eller 5. Noget i den retning (5|[1-9][0-9]*[50]).

2. Nej. Det vil jeg ikke mene. Vi kan ikke på den måde tælle med regulerer udtryk.

3. Som du jo pointerede, så kan man faktisk bare skrive det ud, da vi arbejder med et en afgrænset mænge af heltal.

   ```
   3|4|5|6|7|8|9|12|21|30|33|34|35|36|37|38|39|40|43|44|...|999957|
   999958|999959|999960|999963|999964|999965|999966|999967|999968|
   999969|999970|999973|999974|999975|999976|999977|999978|999979|
   999980|999983|999984|999985|999986|999987|999988|999989|999990|
   999993|999994|999995|999996|999997|999998|999999.
   ```

B)

I Forhold til PL/1 of Fortron så benytter vi os ikke af lookahead operatorer, hvilket vil er krævet for at kunne parse koden i opgaven.