Compiler Construction
Spring 2019

Assignement 2 : Lexical Analysis

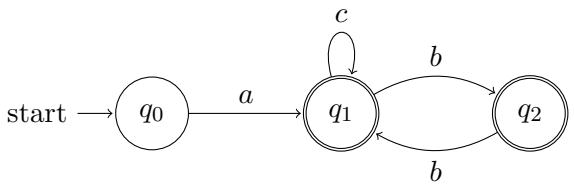
Author : Heinz Hegi, Sylvain Julmy

Prof. Dr. O. Nierstrasz, Dr. Mohammad Ghafari
Manuel Leuenberger, Rathesan Iyadurai

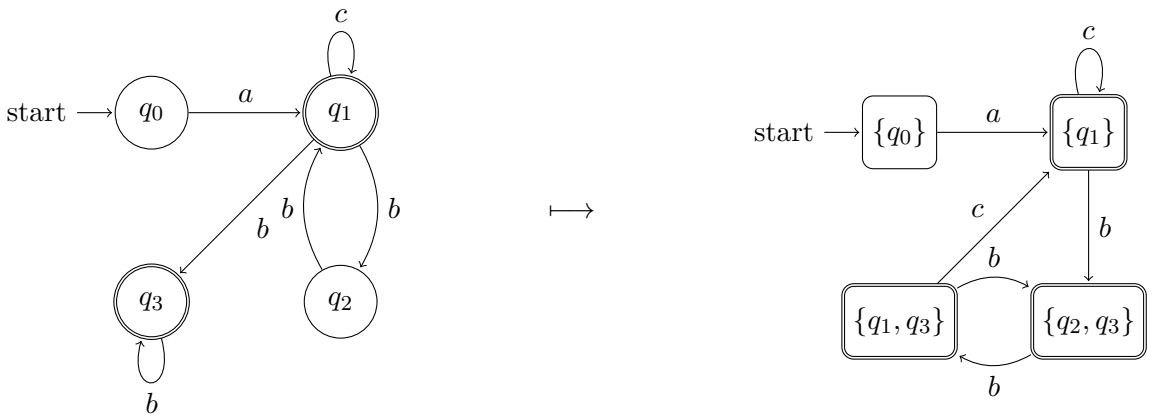
Exercise 1

a)

There exists a very simple solution but it is deterministic :



We also construct a NFA in order to perform the determinization algorithm :



Exercise 2

We are going to show that the language $L = \{a^n b^n | n \geq 0\}$ over the alphabet $\Sigma = \{a, b\}$ is not regular.

Proof. We assume that L is regular, then the Pumping Lemma for regular languages must hold. Let $\omega = a^n b^n$, therefore $x \in L$ and $|x| \leq n$. We have, by the pumping lemma, that $\exists u, v, w$ such that

$$x = uvw \tag{1}$$
$$|uv| \leq n \tag{2}$$
$$|v| \geq 1 \tag{3}$$
$$\forall i \geq 0 : uv^i w \in L \tag{4}$$

We show that $\forall u, v, w$, (1) to (4) don't hold. If (1) to (3) hold, then $x = a^n b^n = uvw$ with $|uv| \leq n$ and $|v| \geq 1$. Therefore $u = a^s$, $v = a^t$, $w = a^p b^n$ with $s + t \leq n$, $t \geq 1$, $p \geq 0$, $s + t + p = n$. But here, (4) don't hold for $i = 0$:

$$uv^0 w = uw = a^s a^p b^n = a^{s+p} b^n \notin L, \text{ since } s + p \neq n$$

□

We can also proof this less formally using the equivalence with finite automatons. The proof is by contradiction : suppose there is a DFA A (possibly converted from a NFA) with k states. It implies that after a^{k+1} steps, we must have visited the same state twice. Let j be the number of steps after which such a doubly visited state has been visited for the first time, and l be the number of steps to the second visit. Therefore, A accepts $a^{j+l} b^j$ if it accepts $a^j b^j$.

Exercise 3

```
b? (an)+ n? a s?
```

Exercise 4

Note that the whitespaces are just for readability. We represent the only actual space in the regex with an underscore. The new lines and additional spaces inside the regular expression aren't sensitive.

Solution which allows multiple spaces :

```
(([1-9][0-9]*d)|
 (((1[0-9])|(2[0-3])|([1-9]))h)|
 ((([1-9])|([1-5][0-9]))(s|m))|
 ([1-9][0-9]?[0-9]?ms)|
 |(_))+
```

Solution in which multiple spaces are not allowed :

```
(([1-9][0-9]*d)                    |
  (((1[0-9])|(2[0-3])|([1-9]))h)   |
  ([1-9][0-9]?[0-9]?ms)            |
  ((([1-9])|([1-5][0-9]))(s|m)))
(_(([1-9][0-9]*d)                  |
  (((1[0-9])|(2[0-3])|([1-9]))h)   |
  ([1-9][0-9]?[0-9]?ms)            |
  ((([1-9])|([1-5][0-9]))(s|m))))*
```