

GTI Übungsblatt 2  
Tutor: Marko Schmellenkamp  
ID: MS1  
Übung: Mi 16-18

Max Springenberg, 177792

## 2.1

**2.1.1** Hannah arbeitet seit einiger Zeit an einem Webportal. Dieses soll nun so erweitert werden, dass Lehrmaterialien in Form komprimierter oder unkomprimierter Archivdateien hochgeladen werden können. Erlaubt sind TAR-Archive, eventuell mit GZIP komprimiert. Gültig sind demnach die Dateiendungen `.tar`, `.tar.gz`, `.tgz`. Der Einfachheit halber sei vorausgesetzt, dass alle Dateinamen ausschließlich Kleinbuchstaben und Punkte beinhalten.

Konstruieren Sie einen  $\epsilon$ -NFA über dem Alphabet  $\Sigma = \{a, \dots, z\}$  mit möglichst wenigen Zuständen, der genau die Dateinamen mit gültigen Dateiendungen akzeptiert.

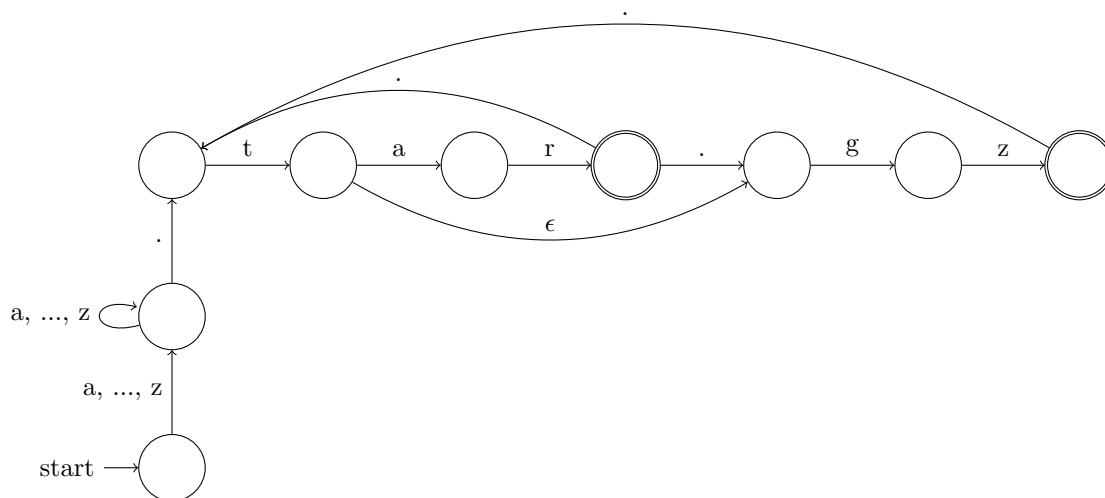
(i) Für alle Wörter der durch den Automaten entschiedenen Sprache  $L$  gilt, dass sie vor der Dateiendung nicht das Zeichen `.` enthalten, aber beliebige Teilwörter aus  $\{a, \dots, z\}^* - \{\epsilon\}$ .

Es wurde nichts zu einer Datei gesagt, die als einziges Teilwort vor der Dateiendung das leere Wort enthält, aber auf Unix-System sind versteckte Dateien durch das Zeichen `.` am Anfang gekennzeichnet. So könnte `.tar` auch eine versteckte Datei mit Namen `tar` ohne Dateiendung sein.

(ii) Ferner muss auf mindestens eine Dateiendung aus der Menge  $\{.tar, .tar.gz, .tgz\}$  geendet werden. Es dürfen keine anderen Dateiendungen (wie z.B. `.zip`) als Teilstrings vorkommen, aber wir nehmen an, dass gueltige Dateiendungen aufeinander folgen dürfen, da mehrfaches verpacken möglich wäre.

Konvention:

Hinsichtlich der Übersichtlichkeit werden Transitionen, wie auch in der Vorlesung, die in den senkenden Zustand führen weggelassen. Falls von einem Zustand aus also die Eingabe eines Zeichens nicht berücksichtigt wird, ist das gleichbedeutend mit einer Transition in den senkenden Zustand.

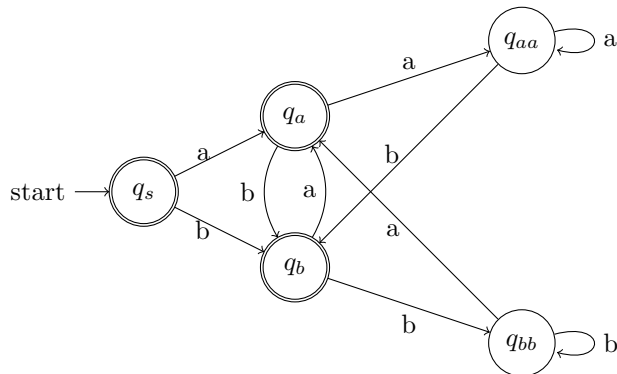


## 2.1.2

Die Aufgabe ist es einen DFA zu konstruieren, der Wörter, die nicht auf `aa` oder `bb` enden akzeptiert. Teilwörter dürfen jedoch auf `aa`, `bb` enden, wenn sie nicht das letzte Teilwort sind.

Zudem soll der Automat nicht mehr als sieben Zustände besitzen.

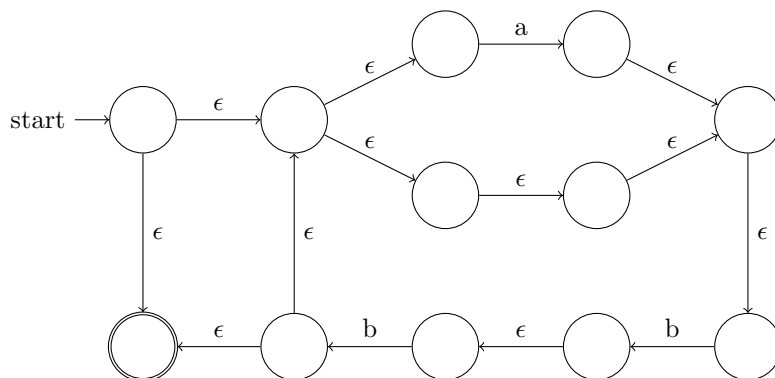
Der Automat:



ist eine Mögliche Lösung, da er nur fünf Zustände besitzt und es gilt:

## 2.2

### 2.2.1



Die Markierten Bereiche entsprechen den jeweiligen Teilwörtern und entsprechen dem Baukastenprinzip aus der Vorlesung, damit ist der Automat entsprechend dem Baukastenprinzip der Vorlesung aufgebaut.

### 2.2.2

Die  $\epsilon$ -closures aller Knoten erbeugen sich zu:

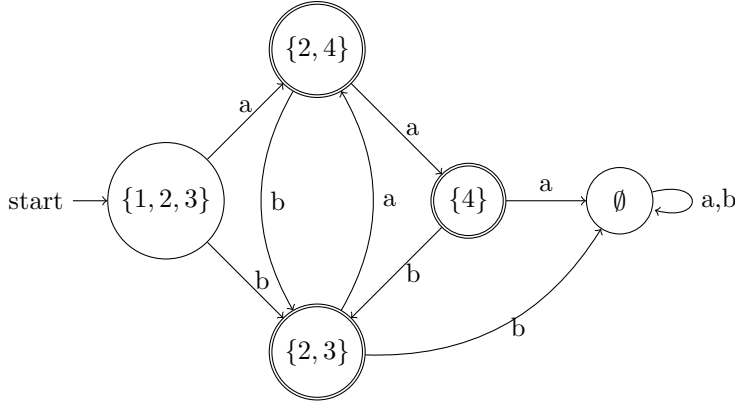
$\epsilon\text{-closure}(1) = \{3, 2\}$ , mit  $1 \xrightarrow{\epsilon} 3 \xrightarrow{\epsilon} 2$

$\epsilon\text{-closure}(2) = \emptyset$ , mit  $\nexists q \in Q : 2 \xrightarrow{\epsilon} q$

$\epsilon\text{-closure}(3) = \{2\}$ , mit  $3 \xrightarrow{\epsilon} 2$

$\epsilon\text{-closure}(4) = \emptyset$ , mit  $\nexists q \in Q : 4 \xrightarrow{\epsilon} q$

Der folgende Automat  $A$ :



ist eine mögliche Lösung.

Der Startzustand umfasst die Menge  $\{1, 2, 3\}$ , da dies der  $\epsilon$ -closure vom Startzustand 1 entspricht. Jeder akzeptierender Zustand ist genau dann ein solcher, wenn er in seiner Menge mindestens einen akzeptierenden Zustand (2 oder 4) enthält.

Die Transitionen von einem Zustand  $q$  bei Eingabe des Zeichens  $\sigma$  ergeben sich über die Konkatenation von  $P_1, \dots, P_n$  mit  $q_k \xrightarrow{\sigma} P_k, 1 \leq k \leq n$  für alle  $q_1, \dots, q_n \in q$ , wobei  $P_k$  die jeweilige Menge von Zuständen die durch Eingabe von  $\sigma$  erreicht werden können, bzw. closure ist.

Ferner gilt, dass dabei auch alle Zustände in der jeweiligen Menge eines Zustands mit den jeweiligen  $\epsilon$ -closures konkateniert wurden und damit auch alle  $\epsilon$ -closures berücksichtigt wurden.

### 2.2.3

Uns interessieren nur die Ausdrücke  $\alpha_{2,2}^2, \alpha_{1,2}^2$ , da 2 der einzige Akzeptierende Zustand ist. wir alhalten  $\alpha_{acc} = \alpha_{2,2}^2 + \alpha_{1,2}^2$  mit:

$$\alpha_{1,1}^0 = a + c$$

$$\alpha_{2,2}^0 = a + b$$

$$\alpha_{1,2}^0 = b$$

$$\alpha_{2,1}^0 = c$$

$$\alpha_{1,1}^1 = \alpha_{1,1}^0 + \alpha_{1,1}^0 (\alpha_{1,1}^0)^* \alpha_{1,1}^0 \equiv (a + c) + (a + c)(a + c)^*(a + c) \equiv (a + c)(a + c)^*$$

$$\alpha_{2,2}^1 = \alpha_{2,2}^0 + \alpha_{2,1}^0 (\alpha_{1,1}^0)^* \alpha_{1,2}^0 \equiv (a + b) + c(a + c)^*b$$

$$\alpha_{1,2}^1 = \alpha_{1,2}^0 + \alpha_{1,1}^0 (\alpha_{1,1}^0)^* \alpha_{1,2}^0 \equiv b + (a + c)(a + c)^*b \equiv (a + c)^*b$$

$$\alpha_{2,1}^1 = \alpha_{2,1}^0 + \alpha_{2,1}^0 (\alpha_{1,1}^0)^* \alpha_{1,1}^0 \equiv c + c(a + c)^*(a + c) \equiv c(a + c)^*$$

$$\alpha_{1,2}^2 = \alpha_{1,2}^1 + \alpha_{1,2}^1 (\alpha_{2,2}^1)^* \alpha_{2,2}^1$$

$$\equiv ((a + c)^*b) + ((a + c)^*b)((a + b) + c(a + c)^*b)^*((a + b) + c(a + c)^*b)$$

$$\equiv (a + c)^*b((a + b) + c(a + c)^*b)^*$$

$$\alpha_{2,2}^2 = \alpha_{2,2}^1 + \alpha_{2,2}^1 (\alpha_{2,2}^1)^* \alpha_{2,2}^1$$

$$\begin{aligned}
&\equiv (a+b) + c(a+c)^*b + ((a+b) + c(a+c)^*b)((a+b) + c(a+c)^*b)^*((a+b) + c(a+c)^*b) \\
&\equiv (a+b) + c(a+c)^*b((a+b) + c(a+c)^*b)^*
\end{aligned}$$

zu:

$$\begin{aligned}
\alpha_{acc} &= (a+c)^*b((a+b) + c(a+c)^*b)^* + (a+b) + c(a+c)^*b((a+b) + c(a+c)^*b)^* \\
&\equiv (a+b) + c^*(a+c)^*b((a+b) + c(a+c)^*b)^*
\end{aligned}$$

## 2.3

Die Sprachen:

$$W_0 = \{w \in \{a,b\}^* \mid |w| \text{ gerade}\}$$

$$W_1 = \{w \in \{a,b\}^* \mid |w| \text{ ungerade}\}$$

$$W_2 = \{w \in \Sigma^* \mid \#_c(w) > 0\}$$

Sind korrekt, da:

Ferner gilt  $L = (W_0 - W_1) - W_2$

Offensichtlich gilt  $(W_0 - W_1) = W_0$ .  $W_0 - W_2$  ergibt sich zu:

$$\{w \in \{a,b\}^* \mid |w| \text{ gerade}\} - \{w \in \Sigma^* \mid \#_c(w) > 0\} = \{w \in \Sigma^* \mid |w| \text{ gerade} \wedge \#_c(w) = 0\} = L$$