

Hochschule Bremerhaven  
University of Applied Sciences

Fakultät II – Management und Informationssysteme

Informatik

Modul Theoretische Informatik

Prof. Dr.-Ing Henrik Lipskoch

**Protokoll zu Aufgabenblatt 02: Team: ti2023\_22**

**Von**

**Ekane Njoh Junior Lesage**

Matrikelnmr: 40128

**Aguiwo II Steve**

Matrikelnmer: 40088

## I. Aufgabe 1

Für diese Aufgabe müssen zwei reguläre Grammatiken konstruiert werden, jeweils eine links und rechte. Dazu haben wir folgendes Beispiel aus dem ersten Übungsblatt verwendet:

```
{"type" : "https://beispiel.com/Junior" , "title" : "You should not pass Ekane." ,  
  "detail" : "Lesage don't give you the permssion to acces this file." ,  
  "instance" : "/account/123/prompt/Njoh"}
```

Für diese Aufgaben mussten wir unsere Grammatikregeln neu anpassen, weil wir sonst mehr als 15 Regeln gehabt hätten.

### a. Mengenangaben

Sei  $G = (\Sigma, V, P, < problem + json >)$

$A = : "https://beispiel.com/Junior" ,$

$B = : "You should not pass Ekane." ,$

$C = : "Lesage don't give you the permssion to acces this file." ,$

$D = : "/account/123/prompt/Njoh" ,$

$T = "type"$

$T2 = "title"$

$De = "detail"$

$It = "instance"$

Seien  $\Sigma = \{ "{"; T ; A ; T2; B ; De ; C ; It ; D ; "}" \}$

Und  $V = \{ < problem + json > ; < type > ; < title > ; < detail > ; < instance > ; < uri > ; < acces > ; < denied > ; < infos > ; < zeichen > \}$

Dann haben wir

### b. rechtsreguläre Grammatik

wir wissen aus [Folie 2 – 1] Satz über reguläre Sprachen, dass sich eine rechtsreguläre Grammatik dadurch auszeichnet, dass links höchstens eine Variable gefolgt von genau einem Buchstaben steht.

$$P = \left\{ \begin{array}{ll} <problem+json> & \rightarrow \{ <type> \\ <type> & \rightarrow T<uri> \\ <uri> & \rightarrow A<title> \\ <title> & \rightarrow T2<denied> \\ <denied> & \rightarrow B<detail> \\ <detail> & \rightarrow De<acces> \\ <acces> & \rightarrow C<instance> \\ <instance> & \rightarrow It<infos> \\ <infos> & \rightarrow D<zeichen> \\ <zeichen> & \rightarrow \} \end{array} \right\}$$

### c. Linksregulär Grammatik

Nach der Konvention der regulären Sprachen besitzt jede rechtsreguläre Grammatik eine linksreguläre Grammatik.

$$P = \left\{ \begin{array}{ll} \langle \text{problem+json} \rangle & \rightarrow \langle \text{zeichen} \rangle \} \\ \langle \text{zeichen} \rangle & \rightarrow \langle \text{infos} \rangle D \\ \langle \text{infos} \rangle & \rightarrow \langle \text{instance} \rangle It \\ \langle \text{instance} \rangle & \rightarrow \langle \text{acces} \rangle C \\ \langle \text{acces} \rangle & \rightarrow \langle \text{detail} \rangle De \\ \langle \text{detail} \rangle & \rightarrow \langle \text{denied} \rangle B \\ \langle \text{denied} \rangle & \rightarrow \langle \text{title} \rangle T2 \\ \langle \text{title} \rangle & \rightarrow \langle \text{uri} \rangle A \\ \langle \text{uri} \rangle & \rightarrow \langle \text{type} \rangle T \\ \langle \text{type} \rangle & \rightarrow \{ \end{array} \right.$$

## II. Aufgabe 2

Bei dieser Aufgabe geht es darum, aus den regulären Sprachen aus der ersten Aufgabe einen (deterministischen) Automaten nach Definition aus der Vorlesung.

Zusätzlich soll auch ein Fehlerzustand für illegale Zeichen und einen Fehlerzustand für legale, aber unpassende Zeichen besitzen. Dabei sollten wir uns passende illegale Zeichen ausdenken.

### a. Automat für die rechtsreguläre Grammatik

Aus [Folie 2 – 5] über deterministischen Automaten wissen wir, Ein (deterministischer) endlicher Automat  $M$  (DEA) ist gegeben durch ein Tupel

Sei  $M = (Z, \Sigma, \delta, z_0, E)$  mit

$Z = \{z_0; z_1; z_2; z_3; z_4; z_5; z_6; z_7; z_8; z_9; z_{10}; z_{11}; z_{12}\}$

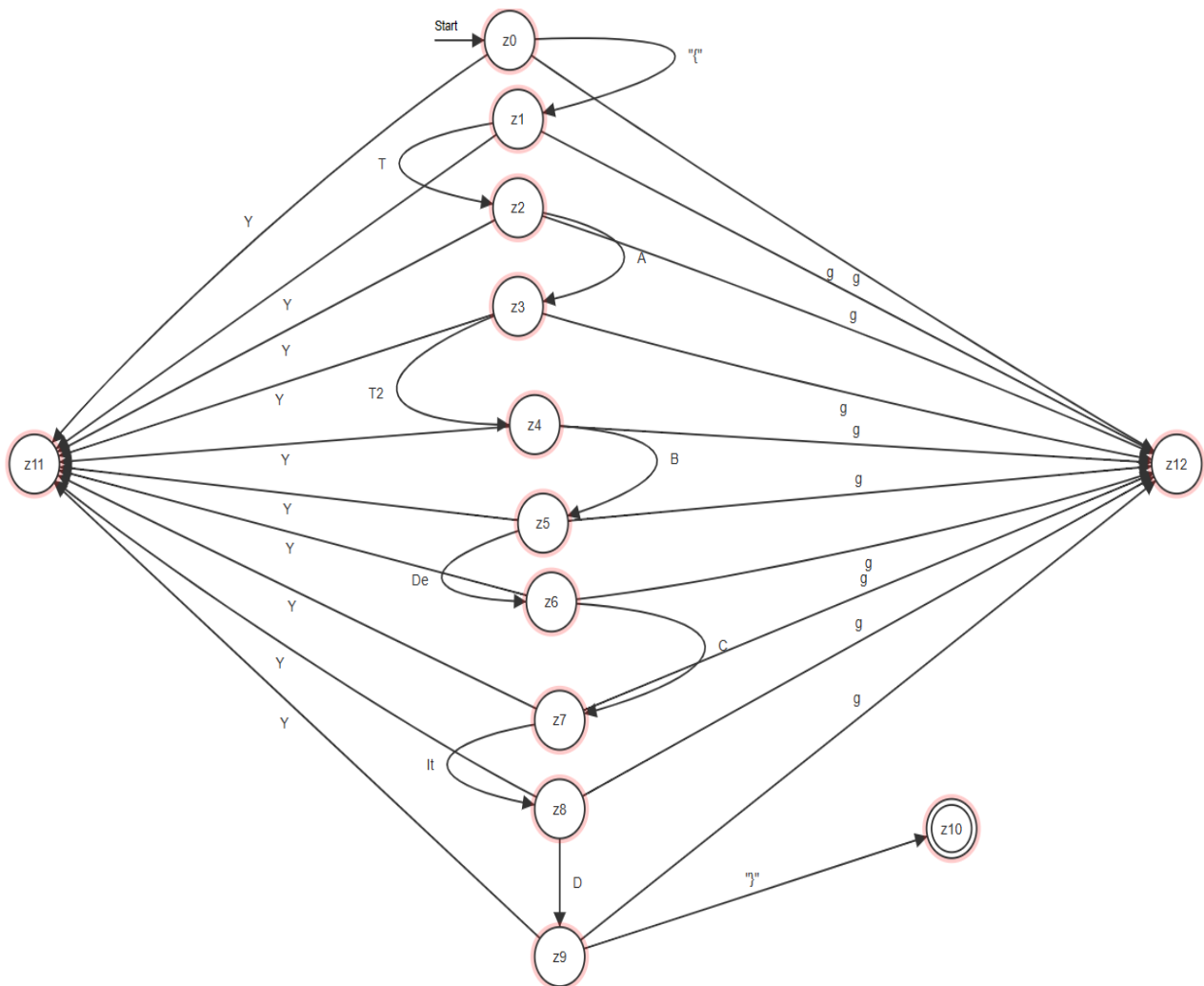
$\Sigma = \{ \{ \}; T; A; T2; B; De; C; It; D; \} \}; g; Y \}$

Seien auch  $g \in G = \{ \#; [; ]; ?; *; \sim; \&; \S; \% ; Moin; !; Bhv \}$ ,

$Y = \text{beliebig erlaubte Eingabe, die für den aktuellen Zustand nicht passt}$

$$\delta = \left\{ \begin{array}{lll} \delta(z0, \{"\}) = z1; & \delta(z4, B) = z5; & \delta(z8, D) = z9; \\ \delta(z0, G) = z12; & \delta(z4, G) = z12; & \delta(z8, G) = z12; \\ \delta(z0, Y) = z11; & \delta(z4, Y) = z11; & \delta(z8, Y) = z11; \\ \delta(z1, T) = z2; & \delta(z5, De) = z6; & \delta(z9, \} ") = z1; \\ \delta(z1, G) = z12; & \delta(z5, G) = z12; & \delta(z9, G) = z1; \\ \delta(z1, Y) = z11; & \delta(z5, Y) = z11; & \delta(z9, Y) = z10; \\ \delta(z2, A) = z3; & \delta(z6, C) = z7; & \delta(z11, G) = z12; \\ \delta(z2, G) = z12; & \delta(z6, G) = z12; & \\ \delta(z2, Y) = z11; & \delta(z6, Y) = z11; & \\ \delta(z3, T2) = z4; & \delta(z7, It) = z8; & \\ \delta(z3, G) = z12; & \delta(z7, G) = z12; & \\ \delta(z3, Y) = z11; & \delta(z7, Y) = z11; & \end{array} \right.$$

$E = \{z10\}$



Bei dieser Aufgabe ist uns aufgefallen, dass es viel leichter war, den Automaten mit der rechtsregulären Grammatik zu konstruieren. Wie haben uns auch die Frage gestellt woran dies liegen könnte und haben uns nach Überlegung geeinigt, dass es sich durch die Ausdruckskraft der linksregulären Grammatik erklären lässt. Ein Automat, der mit der linksregulären Grammatik erzeugt wurde, ist total verkehrt von dem, was man sich im Regelfall vorstellen könnte. Denn das Ende ist dort der Anfang und der Anfang des Ausdrucks stellt das Ende dar.

### Aufgabe 3

## Literaturverzeichnis

<https://www.rfc-editor.org/rfc/rfc7807>

[regulaere-grammatik-endlicher-automat uni-muenster 2008 \(ziemke-koeln.de\)](#)

[Theoretische Informatik \(hs-bremerhaven.de\)](#)

Letzter Zugriff: am 02.11.2023 um 22.26