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

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**](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.

Mengenangaben

rechtsreguläre Grammatik

Sei G = (Σ, V, P, <problem+json>)

A = **: "**[**https://beispiel.com/Junior**](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 Σ = { **"{"; T** ; **A** ; **T2; B ; De ; C ;It ; D ; "}"**} und

V = {<problem+json>; <type>; <title>; <detail>; <instance>; <uri>; <acces> ; <denied>; <infos>; <zeichen> }. Dann haben wir

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

**<problem+json> → {<type>**

**<type> → T<uri>**

**<uri> → A<title>**

**<title> → T2<denied>**

**<denied> → B<detail>**

**<detail> → De<acces>**

**<acces> → C<instance>**

**<instance> → It<infos>**

**<infos> → D<zeichen>**

**<zeichen> → }**

Linksregulär Grammatik

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

**<problem+json> → <zeichen>}**

**<zeichen> → <infos>D**

**<infos> → <instance>It**

**<instance> → <acces>C**

**<acces> → <detail>De**

**<detail> → <denied>B**

**<denied> → <title>T2**

**<title> → <uri>A**

**<uri> → <type>T**

**<type> → {**

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.

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 mit

Z = {z0; z1; z2; z3; z4; z5; z6; z7; z8; z9; z10; z11; z12}

Σ = {**"{"; T**; **A**; **T2; B; De; C; It; D; "}"**}

jORDAN