

Hochschule Bremerhaven  
University of Applied Sciences

Fakultät II – Management und Informationssysteme

Informatik

Modul Theoretische Informatik

Prof. Dr.-Ing Henrik Lipskoch

**Protokoll zu Aufgabenblatt 05: Team: ti2023\_22**

**Von**

**Ekane Njoh Junior Lesage**

Matrikelnmr: 40128

**Aguiwo II Steve**

Matrikelnmer: 40088

## Inhalt

I. Aufgabe 1.....	2
II. Aufgabe 2.....	4
a. H.....	Fehler! Textmarke nicht definiert.
b. H.....	Fehler! Textmarke nicht definiert.
c. H.....	Fehler! Textmarke nicht definiert.
d. H.....	Fehler! Textmarke nicht definiert.
e. H.....	Fehler! Textmarke nicht definiert.
III. Literaturverzeichnis .....	7

### I. Aufgabe 1

Es handelt es sich bei dieser Aufgabe um die Formulierung unseres RFCs als echt-kontextfreie Sprache. Dazu sollte wir in unseren RFC nach einer echt-kontextfreie Struktur zu suchen und dabei folgende Punkte beachten:

- Das Extrahieren der Regeln (begrenzen Sie auf ca. 10-15) für die echt-kontextfreie Struktur
- und Formulierung dieser als eine echt-kontextfreie und zusammenhängende Grammatik
- und zwar mit Regeln in Chomsky-Normalform,
- mit den vereinbarten Symbolen und der Schreibweise aus der Vorlesung auf.

#### Anmerkungen:

- Da unser RFC keine echt-kontextfreie Struktur hat, haben wir uns eine dazu passende echt-kontextfreie Struktur überlegt.
- Wir werden uns bei dieser Aufgabe auf die Produktion eines Pflichtfeldes in unserem RFC begrenzen. Und zwar das Feld „type“.

Aus [Folie] 5 – 8 Chomsky-Normalform wissen wir bereits, dass eine kontextfreie Grammatik  $G = (\Sigma, V, P, S)$  mit  $\epsilon \notin L(G)$  ist in Chomsky-Normalform (CNF) genau dann, wenn alle Regeln aus  $P$ :  
entweder der Form  $X \rightarrow YZ$  oder der Form  $X \rightarrow a$ ,

mit  $X, Y, Z \in V$  und  $a \in \Sigma$  sind.

Deswegen lässt sich Folgendes ableiten:

Es ist  $G = (\Sigma, V, P, problem + json)$  Dabei betrachten wir erstmal die Menge  $V$ , die alle unserer Variablen enthält.

$\Sigma = \{"{" ; "type" ; "https://"; domain; "."com ; "}" \}$

Das Verfahren hierfür ist so gegliedert, dass wir vom BNF zu CNF gelangen

Unsere Regelmenge  $P = \{$

$\langle problem + json \rangle ::= "{" \langle type \rangle "}$   
 $\langle type \rangle ::= \langle required \rangle \langle uri \rangle$   
 $\langle required \rangle ::= "type" :$   
 $\langle uri \rangle ::= "https://" \langle domain \rangle \langle tld \rangle$   
 $\langle domain \rangle ::= domain$   
 $\langle tld \rangle ::= ".com$

$\}$

Wird zu  $P = \{$

$S \rightarrow ABZ$   
 $A \rightarrow \{$   
 $B \rightarrow "type" : D$   
 $D \rightarrow https:// T$   
 $T \rightarrow domain$   
 $G \rightarrow .com$   
 $Z \rightarrow \}$

$\}$

mit  $V = \{S; A; B; Z; D; T; G\}$  und  $\Sigma = \{"{" ; "type" ; "https://"; domain; "."com ; "}" \}$

Es wurden die Variablen in die Form aus der Vorlesung überführt.

Wir haben wir hier leider noch keine echt-kontextfreie Grammatik in der CNF, da unsere Regeln noch nicht der Form  $X \rightarrow YZ$  oder der Form  $X \rightarrow a$  entsprechen, daher ist eine Erweiterung durch den nachfolgenden Schritt nötig.

Wird zu  $P = \{$

1.  $S \rightarrow AB$
2.  $B \rightarrow CD$
3.  $D \rightarrow EF$
4.  $F \rightarrow XZ$
5.  $X \rightarrow OG$
6.  $O \rightarrow TG$

7.  $A \rightarrow \{$
8.  $C \rightarrow \text{"type"} :$
9.  $E \rightarrow \text{https://}$
10.  $T \rightarrow \text{domain.}$
11.  $G \rightarrow \text{com}$
12.  $Z \rightarrow \}$
- $\}$

Es sind inzwischen neue Variablen aufgetaucht, sodass sich unsere Variable Menge geändert hat.

$$V = \{S; A; B; Z; D; T; G; F; X; O; E; C\}$$

Allerdings bleibt die Ausgabe unverändert.

Nun entsprechen unsere neuen Produktionsregeln die CNF, denn jede Regel hat die Form  $X \rightarrow YZ$  oder die Form  $X \rightarrow a$ , wobei A, B und C Nichtterminale und a Terminalsymbole sind. Alle Nichtterminale in den Regeln haben korrekte Ableitungen und es gibt keine Regeln mit leeren Ableitungen.

## II. Aufgabe 2

Diese Aufgabe befasst sich ebenfalls mit echt-kontextfreien Strukturen, wobei wir uns eine Programmiersprache aussuchen, die das Programm *pymmentize* kennt, d.h. eine aus der Liste, die das Programm zeigt, wenn man in der Konsole *pymmentize -L lexer* aufruft.

Zur Lösung der Aufgabe sind wir dem folgenden Weg gegangen:

### a. Ausgangssprache

1. Die von uns gewählte Programmiersprache ist Bash und seine Regeln sehen wie folgt aus:
2.  $\langle \text{bash-script} \rangle ::= \langle \text{greeting} \rangle \langle \text{conditional-execution} \rangle \langle \text{file-operation} \rangle \langle \text{read-input} \rangle$
3.  $\langle \text{greeting} \rangle ::= \text{"echo"} \langle \text{string} \rangle$
4.  $\langle \text{conditional-execution} \rangle ::= \text{"if"} \text{"["} \langle \text{condition} \rangle \text{"}" "then"} \langle \text{bash-script} \rangle \text{"fi"}$
5.  $\langle \text{condition} \rangle ::= \langle \text{comparison} \rangle$
6.  $\langle \text{comparison} \rangle ::= \langle \text{number} \rangle \langle \text{comparison-operator} \rangle \langle \text{number} \rangle$
7.  $\langle \text{comparison-operator} \rangle ::= \text{"-eq"} \mid \text{"-ne"} \mid \text{"-lt"} \mid \text{"-le"} \mid \text{"-gt"} \mid \text{"-ge"}$
8.  $\langle \text{file-operation} \rangle ::= \text{"echo"} \langle \text{string} \rangle \text{">"} \langle \text{filename} \rangle$
9.  $\langle \text{filename} \rangle ::= \langle \text{string} \rangle$
10.  $\langle \text{read-input} \rangle ::= \text{"read"} \text{"-p"} \langle \text{string} \rangle \text{"echo"} \langle \text{string} \rangle$

X

### b. Konvertierung der Sprache in echt-kontext

Wird zu P

1.  $S \rightarrow ABCD$

2.  $A \rightarrow EF$
3.  $B \rightarrow IHKJTS M$
4.  $H \rightarrow [$
5.  $J \rightarrow ]$
6.  $K \rightarrow L$
7.  $L \rightarrow N Q N$
8.  $Q \rightarrow Z$
9.  $C \rightarrow EFUX$
10.  $X \rightarrow F$
11.  $D \rightarrow R V F E F$
12.  $E \rightarrow \text{echo}$
13.  $F \rightarrow \text{string}$
14.  $I \rightarrow \text{if}$
15.  $K \rightarrow \text{condition}$
16.  $T \rightarrow \text{then}$
17.  $M \rightarrow \text{fi}$
18.  $L \rightarrow \text{comparaison}$
19.  $N \rightarrow \text{number}$
20.  $Q \rightarrow \text{comparaison-operator}$
21.  $Z \rightarrow \text{eq}$
22.  $U \rightarrow >$
23.  $X \rightarrow \text{filename}$
24.  $R \rightarrow \text{read}$
25.  $V \rightarrow -p$

Wir werden jetzt unser variablen nummerieren wir haben dann also

$S=A1$	$H=A12$
$A= A2$	$J=A13$
$B= A3$	$K=A14$
$C= A4$	$L=A15$
$D=A5$	$Q=A16$
$H=A6$	$X=A17$
$E=A7$	$F=A18$
$I=A8$	$T=A19$
$M=A9$	$N=A20$
$Q= A10$	$Z=A21$
$U= A11$	$R=A22$
$V=A12$	$F=A23$

Wir wissen aus der Vorlesung [folie] 5-12 (CNF Beweis) , dass  $A_i \rightarrow A_j \alpha \in P$  folgt  $i < j$ .

Daraus bekommen wir dann diesen Ergebniss.

1.  $A_1 \rightarrow A_2 A_3 A_4 A_5$
2.  $A_2 \rightarrow A_7 A_{23}$
3.  $A_3 \rightarrow A_8 A_6 A_{14} A_{13} A_{19} A_1 A_9$
4.  $A_{12} \rightarrow [$
5.  $A_{13} \rightarrow ]$
6.  $A_{14} \rightarrow L$
7.  $A_{15} \rightarrow A_{20} A_{10} A_{20}$
8.  $A_{16} \rightarrow A_{21}$
9.  $A_4 \rightarrow A_7 A_{23} A_{11} A_{17}$
10.  $A_{17} \rightarrow A_{23}$
11.  $A_5 \rightarrow A_{22} A_{12} A_{18} A_7 A_{23}$
12.  $A_7 \rightarrow \text{echo}$
13.  $A_{23} \rightarrow \text{string}$
14.  $A_8 \rightarrow \text{if}$
15.  $A_{14} \rightarrow \text{condition}$
16.  $A_{19} \rightarrow \text{then}$
17.  $A_9 \rightarrow \text{fi}$
18.  $A_{15} \rightarrow \text{comparaison}$
19.  $A_{20} \rightarrow \text{number}$
20.  $A_{10} \rightarrow \text{comparaison-operator}$
21.  $A_{21} \rightarrow \text{eq}$
22.  $A_{11} \rightarrow >$
23.  $A_{17} \rightarrow \text{filename}$
24.  $A_{22} \rightarrow \text{read}$
25.  $A_{12} \rightarrow -p$

Jetzt müssen wir dann unsere Regeln aufbrechen, sodass wir zu dieser Form  $A \rightarrow BC$  oder  $B \rightarrow a$  (wobei a ein Buchstabe ist und A und B sind Variablen)

1.  $A_1 \rightarrow A_2 B_1$
2.  $A_2 \rightarrow A_7 A_{23}$
3.  $A_3 \rightarrow A_8 B_2$
4.  $A_{12} \rightarrow [$
5.  $A_{13} \rightarrow ]$
6.  $A_{14} \rightarrow \text{condition}$
7.  $A_{15} \rightarrow A_{20} B_3$
8.  $A_{16} \rightarrow A_{21}$

9.  $A_4 \rightarrow A_7 B_4$
10.  $A_{17} \rightarrow A_{23}$
11.  $A_5 \rightarrow A_{22} B_5$
12.  $A_7 \rightarrow \text{echo}$
13.  $A_{23} \rightarrow \text{string}$
14.  $A_8 \rightarrow \text{if}$
15.  $A_{14} \rightarrow \text{condition}$
16.  $A_{19} \rightarrow \text{then}$
17.  $A_9 \rightarrow \text{fi}$
18.  $A_{15} \rightarrow \text{comparaison}$
19.  $A_{20} \rightarrow \text{number}$
20.  $A_{10} \rightarrow \text{comparaison-operator}$
21.  $A_{21} \rightarrow \text{eq}$
22.  $A_{11} \rightarrow >$
23.  $A_{17} \rightarrow \text{filename}$
24.  $A_{22} \rightarrow \text{read}$
25.  $A_{12} \rightarrow -p$

Wir haben jetzt unsere Regeln zu der Form  $A \rightarrow BC$  oder  $A \rightarrow a$  (Chomsky-Normal-Form)

Unsere Grammatik wäre dann :

$$\Sigma = \{ [ ; ] , \text{comparaison} ; \text{echo} ; \text{string} ; \text{if} ; \text{condition} ; \text{then} ; \text{fi} ; \text{number} ; \text{comparaison} \\ - \text{operator} ; \text{eq} ; > ; \text{filename} ; \text{read} ; -p \}$$

$$V = \{ A_1 ; A_2 ; A_3 ; \dots ; A_{23} ; B_1 ; B_2 ; \dots B_5 ; \}$$

### III. Literaturverzeichnis

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

[Application error: a client-side exception has occurred \(codecentric.de\)](#)

[JSON - GeeksforGeeks](#)