

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	5
a. H.....	5
b. H.....	5
c. H.....	5
d. H.....	5
e. H.....	5
III. Literaturverzeichnis.....	5

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“.
- Hier werden die Spitzengklammern aus der BNF beibehalten, weil unsere Variable sowie einige unserer Buchstaben aus Zeichenkette bestehen und wir dafür eine eindeutige Identifizierung brauchen.

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:

Unsere Regelmenge P :

$\langle problem + json \rangle \rightarrow \{" \langle type \rangle \}$
 $\langle type \rangle \rightarrow A \langle uri \rangle \langle string \rangle$
 $\langle string \rangle \rightarrow \langle char \rangle | "." | "{" | "}" | "/" | "\" | ":" | "'" | \langle char \rangle \langle string \rangle$
 $\langle char \rangle \rightarrow A|B|C| \dots |Z|a|b|c| \dots |z|0|1|2|3| \dots |9| - |_| \langle char \rangle$
 $\langle uri \rangle \rightarrow "https://" \langle char \rangle "." \langle tld \rangle \langle string \rangle \langle char \rangle$
 $\langle tld \rangle \rightarrow A|B|C| \dots |Z|a|b|c| \dots |z|0|1|2|3| \dots |9| \langle tld \rangle$

Wird zu

1. $\langle problem + json \rangle \rightarrow \langle type \rangle \langle problem + json - erweitert \rangle$
2. $\langle problem + json - erweitert \rangle \rightarrow \langle erstes - zeichen \rangle \langle type \rangle$
3. $\langle erstes - zeichen \rangle \rightarrow \{"$
4. $\langle type \rangle \rightarrow \langle type - erweitert \rangle \langle type - erweitert2 \rangle$
5. $\langle type \rangle \rightarrow A$
6. $\langle type - erweitert \rangle \rightarrow \langle protokoll \rangle \langle uri \rangle$
7. $\langle protokoll \rangle \rightarrow "https://"$
8. $\langle uri \rangle \rightarrow \langle uri0 \rangle \langle letztes - zeichen \rangle$
9. $\langle uri0 \rangle \rightarrow \langle uri1 \rangle \langle uri2 \rangle$
10. $\langle uri1 \rangle \rightarrow \langle char \rangle \langle uri1 \rangle$
11. $\langle char \rangle \rightarrow A|B|C| \dots |Z|a|b|c| \dots |z|0|1|2|3| \dots |9| "." | "{" | "}" | "/" | "\" | ":" | "'" |$
12. $\langle uri2 \rangle \rightarrow \langle tld - char \rangle \langle uri2 \rangle$
13. $\langle tld - char \rangle \rightarrow a|b|c| \dots |z$
14. $\langle letztes - zeichen \rangle \rightarrow \}"$

Es gibt allerdings Regeln, die auf sich selbst verweisen. Daher entspricht das noch nicht wirklich die CNF. Um dieses Problem zu lösen haben wir anhand [Folie] 5 – 12 Algorithmus zur Umformung versucht die Zyklen aufzulösen. Daraus ergibt sich dann Folgendes:

- Auflösen der Zyklen und Umnummerierung der Variablen, sodass aus $(A_i \rightarrow A_j \alpha) \in P$ folgt $i < j$.

Es gibt als Zyklen $\langle uri1 \rangle$ und $\langle uri2 \rangle$

$\langle uri1 \rangle \rightarrow \langle char \rangle \langle uri1 \rangle$ wird zu
 $\langle uri1 \rangle \rightarrow \langle char \rangle \langle uri4 \rangle$
 $\langle uri4 \rangle \rightarrow \langle char \rangle \langle uri4 \rangle \mid \epsilon$
 Und

$\langle uri2 \rangle \rightarrow \langle tld - char \rangle \langle uri2 \rangle$ wird zu
 $\langle uri2 \rangle \rightarrow \langle tld - char \rangle \langle uri5 \rangle$
 $\langle uri5 \rangle \rightarrow \epsilon \mid \langle tld - char \rangle \langle uri5 \rangle$

Auf dieser und weiteren Folien bedeuten

- A_i jeweils eine Variable
- α, β Wortteile aus Variablen und Buchstaben, regulärer Ausdruck $\alpha \equiv (A|t)^*$, d.h. es ist möglich, dass der Wortteile nur aus Buchstaben, nur aus Variablen, aus einem Gemisch von Variablen und Buchstaben besteht oder auch leer ist.

➤ Rückwärtseinsetzen

Es gibt keine Regeln der Form $A \rightarrow B$, daher entfällt dieser Schritt.

➤ Hinzufügen von Buchstabenvariablen

Dieser Schrittentfällt ebenfalls, denn In Regeln der Form $F \rightarrow f$ wird f nicht ersetzt.

- Aufbrechen von langen Regeln

Keine Regel muss in diesem Fall aufgebrochen werden, da keine Regel der Form $A \rightarrow A_1 A_2 A_3 \dots A_k$ existiert.

➤ Schließlich erhalten wir:

1. $\langle \text{problem} + \text{json} \rangle \rightarrow \langle \text{type} \rangle < \text{problem} + \text{json} - \text{erweitert} >$
2. $\langle \text{problem} + \text{json} - \text{erweitert} \rangle \rightarrow \langle \text{erstes} - \text{zeichen} \rangle < \text{type} \rangle$
3. $\langle \text{erstes} - \text{zeichen} \rangle \rightarrow \text{"{"}$
4. $\langle \text{type} \rangle \rightarrow \langle \text{type} - \text{erweitert} \rangle < \text{type} - \text{erweitert2} >$
5. $\langle \text{type} \rangle \rightarrow \alpha$
6. $\langle \text{type} - \text{erweitert} \rangle \rightarrow \langle \text{protokoll} \rangle < \text{uri} \rangle$
7. $\langle \text{protokoll} \rangle \rightarrow \text{"https://"}$
8. $\langle \text{uri} \rangle \rightarrow \langle \text{uri0} \rangle < \text{letztes} - \text{zeichen} \rangle$
9. $\langle \text{uri0} \rangle \rightarrow \langle \text{uri1} \rangle < \text{uri2} \rangle$
10. $\langle \text{uri1} \rangle \rightarrow \langle \text{char} \rangle < \text{uri4} \rangle$
11. $\langle \text{uri4} \rangle \rightarrow \langle \text{char} \rangle < \text{uri4} \rangle \mid \varepsilon$
12. $\langle \text{char} \rangle \rightarrow A|B|C| \dots |Z|a|b|c| \dots |z|0|1|2|3| \dots |9| \text{"."} | \text{"{"} | \text{"}"}$
13. $\langle \text{uri2} \rangle \rightarrow \langle \text{tld} - \text{char} \rangle < \text{uri5} \rangle$
14. $\langle \text{uri5} \rangle \rightarrow \varepsilon \mid \langle \text{tld} - \text{char} \rangle < \text{uri5} \rangle$
15. $\langle \text{tld} - \text{char} \rangle \rightarrow a|b|c| \dots |z$
16. $\langle \text{letztes} - \text{zeichen} \rangle \rightarrow \text{"}"}$

Letztendlich ist unsere Grammatik mit neuen Variablen und Buchstaben befüllt, die auch in einer Menge angegeben werden müssen.

$$\Sigma = \{A; B; C; D; E; F; G; H; I; J; K; L; M; N; O; P; Q; R; S$$

$$; T; U; V; W; X; Y; Z; a; b; c; d; e; f; g; h; i; j; k; l; m; n; o; p; q$$

$$; r; s; t; u; v; w; x; y; z; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9$$

$$; ", " , " - ", " ", "https://", ":", "{", "}", "(", ")", "[]", "\alpha\}$$

$$V = \{ \langle problem + json \rangle; \langle type \rangle; \langle problem + json - erweitert \rangle; \langle erstes - zeichen \rangle; \\ \langle type - erweitert \rangle; \langle type - erweitert2 \rangle; \langle protokoll \rangle; \langle uri \rangle; \langle uri0 \rangle; \\ \langle uri1 \rangle; \langle uri4 \rangle; \langle char \rangle; \langle uri2 \rangle; \langle uri5 \rangle; \langle tld - char \rangle; \langle uri3 \rangle \}$$

II. Aufgabe 2

a. H

b. H

c. H

d. H

e. H

III. Literaturverzeichnis

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

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

[JSON - GeeksforGeeks](#)