Hochschule Bremerhaven
University of Applied Sciences

Fakultät II – Management und Informationssysteme
Informatik
Modul Theoretische Informatik
Prof. Dr.-Ing Henrik Lipskoch

Protokoll zu Aufgabenblatt 09: Team: ti2023_22

Von

Ekane Njoh Junior Lesage Matrikelnmr: 40128
Aguiwo II Steve Matrikelnmer: 40088

Inhalt

I. Aufgabe 1

Bei dieser Aufgabe handelt es sich darum eine Turingmaschine zu definieren, die ein einziges Wort aus unserer Sprache vom Blatt 05 Aufgabe 02 erkennen kann. Dabei sollen wir einiges beachten:

- Die Länge des Wortes soll wenigstens 10 Buchstaben sein.
- Das Wort soll den Wiederholungsteil Ihrer Sprache enthalten (der Teil, warum die Sprache echt kontextfrei ist)
- Bei Erfolg soll die Maschine nach links laufen und den Vornamen eines der Teammitglieder auf das Band schreiben; alle anderen Buchstaben werden zu blank.

a. Wortwahl

Es soll ein Wort gewählt werden, dass mindestens 10 Buchstaben lang ist. Hierfür haben wir folgendes Wort ausgewählt, das 12 Zeichen lang ist:

$$if\ [\ 10\ -eq\ 8\ ];\ then\ echo\ ok\ ;\ fi$$

b. Turingmaschine

Zusammen haben wir folgende Turingmaschine M definiert:

$M\ =\ (Z, \Sigma, \Gamma, \delta, z0, \square, E)$

$Z\ = \{\ z_0; z_1; \dots; z_n; z_s; z_r\}$

z_s ist hierbei der Endzustand, in dem die Maschine den Namen „JUNIOR“ auf das Band schreibt.

z_r stellt der Ablehnungszustand, in den die Maschine übergeht, wenn das Wort nicht den Anforderungen entspricht.

$\Sigma\ =\ \{\ 'if';\ '[';\ '10';\ '-eq';\ '8';\ ']'\ ;\ ';';\ 'then';\ 'echo';\ 'ok';\ ';';\ 'fi'\}$

$\Gamma\ =\ \{\ 'if', '[', '10', '-eq', '8', ']', ';', 'then', 'echo', 'ok', 'fi', '\square', 'J', 'U', 'N', 'I', 'O', 'R', 'Ö', '\%', '\$', '\&', 'ß', 'Ä'\}$

$E = \{\ z_s\ \}$

Das Konzept für die Zustandsüberführungsfunktion δ, beinhaltet spezifische Zustände für jede Phase der Wortanalyse, beginnend mit dem Startzustand, der das Lesen des Wortes initiiert. Jeder Zustand ist zuständig für die Überprüfung bestimmter Wortteile und Strukturen. Die Maschine wechselt Zustände basierend auf der erfolgreichen Erkennung dieser Teile und führt entsprechende Aktionen aus, wie das Schreiben von Markierungen oder die Bewegung des Lese-/Schreibkopfes. Bei Erfüllung aller Bedingungen schreibt die Maschine "JUNIOR" rückwirkend auf dem Band.

$$\delta(z_0, if) = (z_1, if, R)$$

$$\forall\, x \in \Gamma \setminus \{\, if \,\} : \delta(z_0, x) = (z_r, x, R)$$

$$\delta(z_1, [) = (z_2, [, R)$$

$$\forall\, x \in \Gamma \setminus \{\, [\,\} : \delta(z_1, x) = (z_1, if, R)$$

$$\delta(z_2, 10) = (z_3, 10, R)$$

$$\forall\, x \in \Gamma\ \{\, 10 \,\} : \delta(z_2, x) = (z_r, x, R)$$

$$\delta(z_3, -ge) = (z_4, -ge, R)$$

$$\forall\, x \in \Gamma\ \{-ge \,\} : \delta(z_3, x) = (z_r, x, R)$$

$$\delta(z_4, 8) = (z_5, 8, R)$$

$$\forall\, x \in \Gamma\ \{\, 8 \,\} : \delta(z_4, x) = (z_r, x, R)$$

$$\delta(z_5,]) = (z_6,], R)$$

$$\forall\, x \in \Gamma\ \{\,] \,\} : \delta(z_5, x) = (z_r, x, R)$$

$$\delta(z_6, ;) = (z_7, ;, R)$$

$$\forall\, x \in \Gamma\ \{\, ; \,\} : \delta(z_6, x) = (z_r, x, R)$$

$$\delta(z_7, then) = (z_8, then, R)$$

$$\forall\, x \in \Gamma\ \{\, then \,\} : \delta(z_7, x) = (z_r, x, R)$$

$$\delta(z_8, echo) = (z_9, echo, R)$$

$$\forall\, x \in \Gamma\ \{\, echo \,\} : \delta(z_8, x) = (z_r, x, R)$$

$$\delta(z_9, ok) = (z_{10}, ok, R)$$

$$\forall\, x \in \Gamma\ \{\, ok \,\} : \delta(z_9, x) = (z_r, x, R)$$

$$\delta(z_{10}, ;) = (z_{11}, ;, R)$$

$$\forall\, x \in \Gamma\ \{\, ; \,\} : \delta(z_{10}, x) = (z_r, x, R)$$

$$\delta(z_{11}, fi) = (z_{12}, fi, R)$$

$$\delta(z_{12}, \square) = (z_r, \square, L)$$

$$\delta(z_r, \text{Ö}) = (z_r, R, L)$$

$$\delta(z_r, \text{Ä}) = (z_r, O, L)$$

$$\delta(z_r, \text{ß}) = (z_r, I, L)$$

$$\delta(z_r, \&) = (z_r, N, L)$$

$$\delta(z_r, \$) = (z_r, U, L)$$

$$\delta(z_r, \%\$\&\text{ß}\,\text{Ä}) = (z_r, J, L)$$

$$\forall\, x \in \Sigma : \delta(z_r, x) = (z_r, \square, L)$$

$$\delta(z_r, \square) = (z_e, \square, R)$$

Aufgrund der Aufforderung der Aufgabe für die Wortakzeptanz der Maschine (ein einziges Wort), haben wir die Maschine sehr eingeschränkt gestaltet.

II. Aufgabe 02

Bei dieser Aufgabe müssen wir Konfigurationsüberleitungen für unsere vorher definierten Maschine angeben. Dies sollen wir sowohl für das von uns gewählte Wort als auch für ein beliebiges anderes Wort mit ein paar Buchstabenfehler machen.

a. Konfigurationsüberleitungen für unser Wort

$\vdash z_0, \boldsymbol{if}\ [\ \mathbf{10}\ -\boldsymbol{ge}\ \mathbf{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \square\square\square\square\square\square\square\square$

$\vdash z_1, [\ \mathbf{10}\ -\boldsymbol{ge}\ \mathbf{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\square\square\square\square\square\square\square$

$\vdash z_2, \mathbf{10}\ -\boldsymbol{ge}\ \mathbf{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\square\square\square\square\square\square\square$

$\vdash z_3, -\boldsymbol{ge}\ \mathbf{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\ \mathbf{10}\square\square\square\square\square\square$

$\vdash z_4, \mathbf{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\ \mathbf{10}\ -\boldsymbol{ge}\square\square\square\square\square$

$\vdash z_5,];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\ \mathbf{10}\ -\boldsymbol{ge}\ \mathbf{8}\square\square\square\square$

$\vdash z_6, \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\ \mathbf{10}\ -\boldsymbol{ge}\ \mathbf{8}\ ];\square\square\square$

$\vdash z_7, \boldsymbol{echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\ \mathbf{10}\ -\boldsymbol{ge}\ \mathbf{8}\ ];\ \boldsymbol{then}\square\square$

$\vdash z_8, \boldsymbol{ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\ \mathbf{10}\ -\boldsymbol{ge}\ \mathbf{8}\ ];\ \boldsymbol{then\ echo}\square$

$\vdash z_9, ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\ \mathbf{10}\ -\boldsymbol{ge}\ \mathbf{8}\ ];\ \boldsymbol{then\ echo\ ok}\square$

$\vdash z_{10}, \boldsymbol{fi}, \boldsymbol{if}\ [\ \mathbf{10}\ -\boldsymbol{ge}\ \mathbf{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\square$

$\vdash z_{11}, \square, \boldsymbol{if}\ [\ \mathbf{10}\ -\boldsymbol{ge}\ \mathbf{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}\square$

$\vdash z_{12}, \square, \boldsymbol{JUNIOR}\square\square\square\square\square\square\square\square\square$

$\vdash z_r, \square, \boldsymbol{JUNIO}\square\square\square\square\square\square\square\square$

$\vdash z_r, \square, \boldsymbol{JUNI}\square\square\square\square\square\square\square$

$\vdash z_r, \square, \boldsymbol{JUN}\square\square\square\square\square\square\square$

$\vdash z_r, \square, \boldsymbol{JU}\square\square\square\square\square\square\square$

$\vdash z_r, \square, \boldsymbol{J}\square\square\square\square\square\square\square\square$

$\vdash z_e, \square, \square\square\square\square\square\square\square\square\square$

b. Konfiguration für ein Wort mit Schreibfehlern

Um den zweiten Teil der Aufgabe für ein Wort mit einigen Buchstabenfehlern zu lösen, gehen wir folgendermaßen vor:

Angenommen, das Wort mit Fehlern ist "$if\ [\ 1x\ -ge\ 8\ ];\ then\ echo\ ok\ ;\ fi$". Die Turingmaschine würde zunächst "if" und "[" korrekt lesen und in die entsprechenden Zustände z1 und z2 übergehen. Bei "1x" statt "10" würde jedoch ein Fehler auftreten:

$\vdash \boldsymbol{z0}, \boldsymbol{if}\ [\ \boldsymbol{1x}\ -\boldsymbol{ge}\ \boldsymbol{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \square\square\square\square\square\square\square\square$

$\vdash \boldsymbol{z1}, [\ \boldsymbol{1x}\ -\boldsymbol{ge}\ \boldsymbol{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\square\square\square\square\square\square\square$

$\vdash \boldsymbol{z2}, \boldsymbol{1x}\ -\boldsymbol{ge}\ \boldsymbol{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ \square\square\square\square\square\square\square$

"[" bestätigt, "1x" anstelle von "10" gelesen, Fehler, geht zu zr.

An diesem Punkt angekommen, wo sie einen Fehler erkennt, da "1x" nicht dem erwarteten Wort "10" entspricht. Daher wechselt sie in den Fehlerbehandlungszustand zr:

$\vdash \boldsymbol{zr}, \boldsymbol{1x}\ -\boldsymbol{ge}\ \boldsymbol{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\square\square\square\square\square\square\square$

Falsches Wort "1x" gelesen, Maschine ist jetzt im Zustand zr und beginnt den Fehlerbehandlungsprozess.

$\vdash \boldsymbol{zr}, -\boldsymbol{ge}\ \boldsymbol{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\boldsymbol{1x}\square\square\square\square\square\square$

Maschine verarbeitet weiter im Fehlerbehandlungszustand.

$\vdash \boldsymbol{zr}, \boldsymbol{8}\ ];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\boldsymbol{1x}\ -\boldsymbol{ge}\square\square\square\square\square$

Weiterhin im Fehlerbehandlungszustand.

$\vdash \boldsymbol{zr},];\ \boldsymbol{then\ echo\ ok}\ ;\ \boldsymbol{fi}, \boldsymbol{if}\ [\boldsymbol{1x}\ -\boldsymbol{ge}\ \boldsymbol{8}\square\square\square\square$

Maschine bleibt im Fehlerbehandlungszustand.

Die Maschine würde in diesem Fehlerbehandlungszustand bleiben, da sie keine passenden Übergänge für das fehlerhafte Wort hat. Sie würde entweder weiterhin den Rest des Wortes lesen, ohne eine sinnvolle Aktion auszuführen, oder anhalten, wenn keine weiteren Übergänge definiert sind.

c. Antwort zu Frage a

Für den endlich oft wiederholten Teil müsste die Maschine einen Mechanismus haben, um zu zählen, wie oft der Wiederholungsteil auftritt. Das könnte durch Zustände erreicht werden, die speziell für das Zählen und Überprüfen der Wiederholungen zuständig sind.

d. Antwort zu Frage b

Für den beliebig oft wiederholten Teil müsste die Maschine in der Lage sein, eine unbestimmte Anzahl von Wiederholungen zu verarbeiten. Das erfordert eine flexiblere Zustandsstruktur, möglicherweise mit einer Schleifenlogik, die es der Maschine erlaubt, den Wiederholungsteil beliebig oft zu durchlaufen und dabei die Übereinstimmung der Anfänge und Enden zu überprüfen.

III. Aufgabe 03

a. Gesucht

Es soll eine Turingmaschine definiiert werden , die wörter unsere Sprache von Aaufgabe 08 erkennen kann. Diese sollen mindestens 10 Buchstaben lang sein und den echt kontextsensitven Teil unserer Sprache enthalten . Anschließend sollen drei Beispiel-Konfigurationsüberleitungen der Maschine gezeigt werden

b. Idee

Sei L = { $eine\ Permutation\ von\ a\ ib\ j\ c\ k : i > j > k \geq 1, i,j, k \in \mathbb{N}$} unsere Sprache aus der Aufgabe 8 .

Unsere Sprache könnte als kontextsensitiv betrachtet, weil wir eine Regel hinzugefügt haben, die die Anzahl der Symbole in einer spezifischen Reihenfolge einschränkt. Die Regel i>j>k definiert, dass die Anzahl der ‚a's größer sein soll, als die Anzahl der ‚b's, die auch größer sein soll als die Anzahl der ‚c's.

Unsere Produktionsregeln sind dann also:

G = (Σ, V, P, S) eine mögliche Grammatik für unsere Sprache würde aus Folgendem bestehen: • Ein Alphabet: Σ = {a; b; c}

• Eine Variablenmenge: V = { S; A; B; X}

Produktionsregeln:

$S \rightarrow aSBC$

$CB \rightarrow AB$

$SB \rightarrow aX$

$X \rightarrow b$

Wir haben die Turingmaschine M defniert:

M = (Z, Σ, Γ, δ, z0, E).

Z = {z0 ,z1 ,z2 ,z3 ,z4 , z ak ,z ab }

Σ = {a,b,c}

Γ = {a,b,c, □ ,X}

• z0 (Anfangszustand)

• z1 (Zustand zum Lesen von S)

• z2 (Zustand zum Lesen von CB)

• z3 (Zustand zum Lesen von SB)

• z4 (Zustand zum Lesen von X)

•Z ak (Akzeptierzustand)

•Z ab (Ablehnungszustand)

Die Funktion δ gibt an, wie sich die Turingmaschine in einem bestimmten Zustand verhält, wenn sie ein bestimmtes Symbol auf dem Band liest. Hier sind die wichtigsten Übergangsregeln:

Wenn die Maschine a im Zustand z0 liest, schreibt sie X und geht nach rechts in den Zustand z1 .

Die Maschine liest b im Zustand z1 , schreibt □ und geht nach links in den Zustand z2 .

Die Maschine folgt dann einer Sequenz von Übergängen, um sicherzustellen, dass die Anzahl der a's größer ist als die Anzahl der b's, und die Anzahl der b's größer ist als die Anzahl der c's.

Am Ende geht die Maschine nach rechts in den Zustand Z ak und akzeptiert das Wort.

δ (z0 ,a)→ (z1 ,X,R) (Lesen von S)

δ (z1 ,b)→(z2 , □,L) (Lesen von CB)

δ (z2 ,a)→(z3 , □,L) (Lesen von SB)

δ (z3 ,a)→(z3 ,a,L) (Ignorieren von a in SB)

δ (z3 ,b)→ (z3 ,b,L) (Ignorieren von b in SB)

δ (z3 , □) → (z4 , □,R) (Lesen von X)

δ (z4 ,b) →(z ak , □,R) (Akzeptieren, wenn b gelesen wird)

δ (z0 ,b) → (z ab , □,R)(Ablehnen von zu kurzen Wörtern)

δ (z1 , □)→(z ab , □,R) (Ablehnen von Wörtern, die nicht mit S δ beginnen)

δ (z2 , □) →(z ab , □,L)(Ablehnen von Wörtern ohne CB)

δ (z3 ,c) → (z ab, □,L) (Ablehnen von Wörtern, die die Bedingung i > j > k nicht erfüllen)

Jetzt kommen wir zu unseren Konfigurationsüberleitungen von Start-zustand bis zum Eerreichen des Endzustandes.

Lassen wir uns nun die Konfigurationsüberleitungen für ein bestimmtes Wort , ZB.

abcbabcbab

⊢ z0 , "aaabbcc", □□□□□□□

⊢ z1 , "Xaabbcc", □□□□□□

⊢ z1, "XXabbcc", □□□□□

⊢ z1, "XXXbbcc", □□□□

⊢ z2 , "XXXBBcc", □□

⊢ z3 , XXXBbcc", □□□

⊢ Z ak, "XXXBBCC", □

Die Maschine hat das Wort akzeptiert und befindet sich im Akzeptierzustand.

Die Turingmaschine startet im Zustand z0 mit dem Wort "aaabbcc", ersetzt in den Zuständen Z1 und z2 sukzessive 'a's und 'b's, verarbeitet in z3 die 'c's und erreicht schließlich den Akzeptierzustand Z ak, nachdem alle Symbole gemäß den Kriterien der Sprache korrekt verarbeitet wurden, was zur Annehmung des Wortes führt.

Die Sprache L, definiert durch die Regel

i>j>k für die Buchstaben 'a', 'b' und 'c', ist ein Beispiel für eine kontextsensitive Sprache. Diese Art von Sprache geht über die Grenzen einer kontextfreien Grammatik hinaus, was sich in der Komplexität der erforderlichen Turingmaschine widerspiegelt.

IV. Literaturverzeichnis

https://elli.hs-bremerhaven.de/goto.php?target=file_338582_download&client_id=elli

Letzter Zugriff am 11.01.2024 um 23 Uhr

https://turingmachinesimulator.com/

Letzter Zugriff am 11.01.00:21 Uhr