TI 1 - Ergänzung 11

A1

Für Sprachen A, B mit B vom Typ k gilt:

$$A \setminus B \text{ ist vom Typ } k \iff A \cup B \text{ ist vom Typ } k.$$

1. Ja! Wir zeigen beide Richtungen der Äquivalenz mithilfe der Abschlusseigenschaften von regulären Sprachen.

„$\Rightarrow$":

Sei $A \setminus B$ regulär (also vom Typ 3).

$\Rightarrow (A \setminus B) \cup B = A \cup B$ regulär.

„$\Leftarrow$":

Sei $A \cup B$ regulär.

$\Rightarrow (A \cup B) \cap \overline{B} = A \setminus B$ $(= A \cap \overline{B})$ regulär, da B und somit auch $\overline{B}$ regulär. $\square$

2. Nein! Weil die Klasse der kontextfreien Sprachen unter Vereinigung abgeschlossen ist, gilt zwar die Richtung „$\Rightarrow$", aber die Rückrichtung kann mit einem Gegenbeispiel widerlegt werden.

Gegenbeispiel:

Für $A = \{a^k b^l c^m \mid k = l\}$ und $B = \{a^k b^l c^m \mid l \neq m\}$ gilt:

- B ist kontextfrei (also vom Typ 2), denn es gilt $B = L(G_B)$ für $G_B = (\{S,T,U,V\}, \{a,b,c\}, P, S)$ mit

$$P = \{ S \to TU \mid TV \mid U \mid V,$$
$$T \to aT \mid a,$$
$$U \to bUc \mid bU \mid b,$$
$$V \to bVc \mid Vc \mid c \}.$$

- A ist kontextfrei, denn es gilt $A = L(G_A)$ für $G_A = (\{S,T,U\}, \{a,b,c\}, P, S)$ mit

$$P = \{ S \to TU \mid T \mid U \mid \varepsilon,$$
$$T \to aTb \mid ab,$$
$$U \to cU \mid c \}.$$

- Da A und B kontextfrei sind, ist auch $A \cup B$ kontextfrei.

- $A \setminus B = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ ist aber nicht kontextfrei.

<u>Bemerkung</u>:

Für $k=1$ lässt sich die Aussage ganz analog zum Fall $k=3$ beweisen. Um die Aussage für $k=0$ zu widerlegen, benötigen wir Sprachen, von denen wir wissen, dass sie nicht vom Typ 0 sind. Dies ist Bestandteil der TI2-Vorlesung.

<u>Nachtrag</u>: Ein weiteres Gegenbeispiel zu Teilaufgabe 2 ist $A = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ und $B = \{a^k b^l c^m \mid k \neq l \vee l \neq m\}$. Dann ist $A \cup B = L(a^* b^* c^*)$ kontextfrei, aber $A \setminus B = A$ nicht.

A2

1. Idee:

Die DTM M überprüft zuerst, ob die Reihenfolge der Zeichen im Eingabewort w korrekt ist, d.h. ob w die Form $a^k b^l c^m$ hat. Hierfür verwendet sie die Zustände 0, 1 und 2.

Danach entfernt sie abwechselnd as, bs und cs bis entweder das gesamte Wort entfernt wurde oder einzelne Buchstaben übrig bleiben. Im ersten Fall akzeptiert sie das Wort (Zustand 7) und im zweiten nicht (dann „bleibt sie stehen"). Für den Entfernungsprozess verwendet sie die Zustände 3 bis 6.

Um leichter erkennen zu können, ob man mit dem Leseknopf das Wort von links oder rechts verlassen hat, entfernen wir Buchstaben, indem wir sie durch x und nicht durch □ ersetzen.

Konstruktion:

Wir definieren $M = (\underbrace{\{0,\dots,7\}}_{\text{Zustände}}, \underbrace{\{a,b,c\}}_{\text{Eingabealphabet}}, \underbrace{\{a,b,c,x,\square\}}_{\text{Bandalphabet}}, \delta, \underbrace{0}_{\text{Startzustand}}, \underbrace{\square}_{\text{Leersymbol}}, \underbrace{\{7\}}_{\text{Endzustände}})$ mit δ wie folgt:

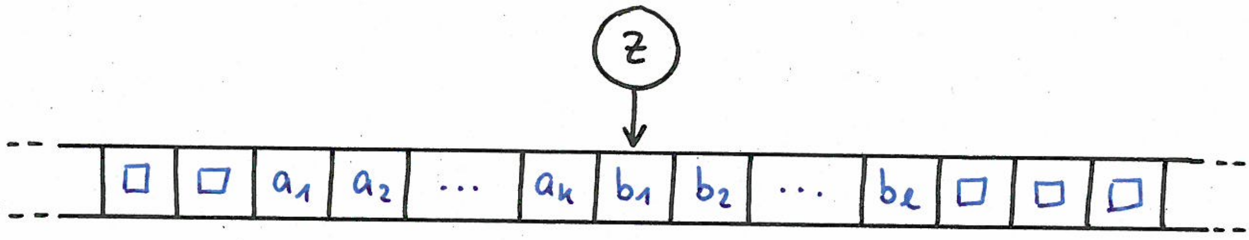
z	$\delta(z,a)$	$\delta(z,b)$	$\delta(z,c)$	$\delta(z,x)$	$\delta(z,\square)$	Intuition
0	$(0,a,R)$	$(1,b,R)$	-	-	$(3,\square,L)$	Startzustand. Lese as am Anfang.
1	-	$(1,b,R)$	$(2,c,R)$	-	$(3,\square,L)$	Lese bs nach den as.
2	-	-	$(2,c,R)$	-	$(3,\square,L)$	Lese cs nach den bs.
3	$(3,a,L)$	$(3,b,L)$	$(3,c,L)$	$(3,x,L)$	$(4,\square,R)$	Bewege Lesekopf zum linken Ende.
4	$(5,x,R)$	-	-	$(4,x,R)$	$(7,\square,N)$	Entferne das linkeste a.
5	$(5,a,R)$	$(6,x,R)$	-	$(5,x,R)$	-	Entferne das linkeste b.
6	-	$(6,b,R)$	$(3,x,L)$	$(6,x,R)$	-	Entferne das linkeste c.
7	-	-	-	-	-	Endzustand.

M soll „stehen bleiben"

Für die restlichen Fälle (die freien Einträge in der Tabelle) definieren wir $\boxed{\delta(z,y) = (z,y,N)}$ für $z \in \{0,\dots,7\}$ und $y \in \{a,b,c,x,\square\}$.

Konfigurationen:

Konfigurationen stellen „Momentaufnahmen" der TM dar. Die Konfiguration $a_1 a_2 \dots a_k z b_1 b_2 \dots b_\ell$ steht für:



Konfigurationsfolge für w=abc:

$$0abc \vdash a0bc \vdash ab1c \vdash abc2 \vdash ab3c \vdash a3bc \vdash 3abc$$
$$\vdash 3\square abc \vdash 4abc \vdash x5bc \vdash xx6c \vdash x3xx \vdash 3xxx$$
$$\vdash 3\square xxx \vdash 4xxx \vdash x4xx \vdash xx4x \vdash xxx4 \vdash xxx7$$

2. Idee:

Als Erstes soll das allererste Zeichen mit dem ersten Zeichen der zweiten Hälfte verglichen werden. Sind sie gleich, so werden sie entfernt (wieder mit x statt □). Diesen Prozess wiederholt man mit dem zweiten, dritten, ... Zeichen beider Worthälften.

Um das erste Zeichen der zweiten Hälfte zu finden, machen wir Gebrauch vom Nichtdeterminismus.

Die (diesmal nichtdeterministische) Turingmaschine M entfernt das allererste Zeichen (Zustand 0) und „merkt sich" dieses (Zustände 1 und 2). Danach „rät" sie das erste Zeichen der zweiten Worthälfte und entfernt es. Dies tut sie, indem sie das Wort von links nach rechts durchgeht und jedes Zeichen, das dem eben entfernten gleicht, sowohl entfernt als auch stehen lässt.

Konstruktion:

Wir definieren $M = (\{0, \dots, 9\}, \{a,b\}, \{a,b,x,\square\}, \delta, 0, \square, \{9\})$ mit δ wie folgt:

z	$\delta(z,a)$	$\delta(z,b)$	$\delta(z,x)$	$\delta(z,\square)$	Intuition
0	$\{(1,x,R)\}$	$\{(2,x,R)\}$	$\emptyset$	$\{(9,\square,N)\}$	Startzustand.
1	$\{(1,a,R),\ (3,x,N)\}$	$\{(1,b,R)\}$	$\emptyset$	$\emptyset$	Entferne erstes Zeichen der zweiten Worthälfte, falls dieses mit dem ersten übereinstimmt.
2	$\{(2,a,R)\}$	$\{(2,b,R),\ (3,x,N)\}$	$\emptyset$	$\emptyset$	
3	$\{(3,a,L)\}$	$\{(3,b,L)\}$	$\{(3,x,L)\}$	$\{(4,\square,R)\}$	Bewege Lesekopf zum linken Ende.
4	$\{(5,x,R)\}$	$\{(6,x,R)\}$	$\{(4,x,R)\}$	$\{(9,\square,N)\}$	Entferne das linkeste a oder b.
5	$\{(5,a,R)\}$	$\{(5,b,R)\}$	$\{(7,x,N)\}$	$\emptyset$	Bewege Lesekopf zur zweiten Worthälfte.
6	$\{(6,a,R)\}$	$\{(6,b,R)\}$	$\{(8,x,N)\}$	$\emptyset$	
7	$\{(3,x,N)\}$	$\emptyset$	$\{(7,x,R)\}$	$\emptyset$	Entferne das linkeste a bzw. b der zweiten Worthälfte, falls das Zeichen mit dem zuletzt entfernten übereinstimmt.
8	$\emptyset$	$\{(3,x,N)\}$	$\{(8,x,R)\}$	$\emptyset$	
9	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	Endzustand.

<u>Bemerkung</u>: Bei NTMs liefert die Überführungsfunktion δ eine Menge von Tripeln. Bei DTMs liefert sie dagegen genau ein Tripel.

Beispiel:

Für das Wort $w = abababab$ ergibt sich der folgende Berechnungsbaum:

$0abababab$

$\top$

$x1bababab$

$\top$

$xb1ababab \vdash xb3xbabab \vdash \dots$

$\top$

$xba1babab$

$\top$

$xbab1abab \vdash xbab3xbab \vdash \dots$

$\top$

$xbaba1bab$

$\top$

$xbabab1ab \vdash xbabab3xb \vdash \dots$

$\top$

$xbababa1b$

$\top$

$xbababab1$

Von dieser Konfiguration aus wird eine akzeptierende Konfiguration erreicht.

Von diesen zwei aus nicht.

3. Bitte selber ausprobieren und ggf. simulieren (siehe Bemerkung auf dem Übungsblatt).