Karlsruher Institut für Technologie
Institute of Information Security and Dependability
Dr. Mattias Ulbrich
https://formal.kastel.kit.edu/

Übungsblatt 11

Grundbegriffe der Informatik — Winter 2023/24

Tutor*in: | Tutorium Nr.:

Nach-,Vorname: ,

Matr.nr.:

Ausgabe: 30. Januar 2023, 14:30 Uhr

Abgabe: 9. Februar 2024, 12:30 Uhr

Bitte beachten Sie die Hinweise auf der letzten Seite.

---

*Von Tutor*in auszufüllen:*

Blatt 11: / 20

Blätter 1 – 11: / 221

Aufgabe 1 - Bonsai (5 Punkte)

Gegeben sei der reguläre Ausdruck R = `(b|∅)a(b*)`.

a) Geben Sie den Kantorowitsch-Baum für den regulären Ausdruck R an. (1 Punkt)

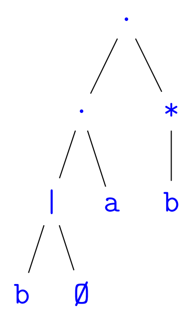
b) Geben Sie eine rechtslineare Grammatik G an, so dass $L(G) = L(R)$. (2 Punkte)

c) Geben Sie **einen** Ableitungsbaum zum Wort `babbb` bezüglich der Produktionen von G an. (1 Punkt)

d) Geben Sie die von dem regulären Ausdruck R' = `(b|∅*)a(b*)` beschriebene Sprache $L(R')$ an. (1 Punkt)

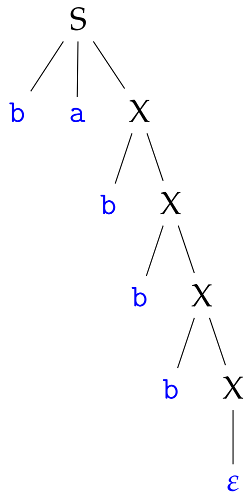
Lösung 1

a) Ein möglicher Kantorowitsch-Baum ist der folgende:



b) $G = (\{S, X\}, \{\mathtt{a}, \mathtt{b}\}, S, P)$, mit $P = \{S \to \mathtt{ba}X,\ X \to \mathtt{b}X \mid \varepsilon\}$

c) Ein möglicher Ableitungsbaum ist der folgende:



d) $L(R') = \{\mathtt{bab}^n, \mathtt{ab}^n \mid n \in \mathbb{N}_0\}$

Aufgabe 2 - Turingmaschine (7 Punkte)

Gegeben sei das Alphabet $A = \{0, 1, \#\}$. In Abbildung 1 finden Sie eine graphische Darstellung der Turingmaschine $T = (Z, z_0, X, f, g, m)$ mit dem Bandalphabet $X = A \cup \{\Box, \$\}$.

a) Führen Sie T für die Eingabe $w = \mathtt{1000\#1011}$ aus und geben Sie jeweils das Arbeitsband an,

- wenn T in den Zustand z_0 übergegangen ist und
- nachdem T gehalten hat (Endkonfiguration).

Markieren Sie jeweils die Position des Lesekopfs durch Einkreisen der Stelle auf dem Band.

Nutzen Sie dazu das Raster aus Abbildung 2. (Es werden nicht alle Zeilen des Rasters benötigt.) (2 Punkte)

b) Geben Sie die Funktion $f_T : \bigcup_{i=0}^{\infty} \left(\{0, 1\}^i \times \{0, 1\}^i\right) \to \{0, 1\}^*$ an, so dass bei Eingabe eines Wortes $w = w_1\#w_2$ mit $w_1, w_2 \in \{0, 1\}^k$ für $k \in \mathbb{N}_0$ nach Anhalten von T das Wort $f_T(w_1, w_2)$ auf dem Band steht. (2 Punkte)

c) Geben Sie Abschätzungen für $Time_T(n)$ und $Space_T(n)$ an, wobei hier n die Länge des Eingabewortes darstellt. In welchen Komplexitätsklassen liegen diese jeweils? Begründen Sie Ihre Antwort. (3 Punkte)

Lösung 2

a)

…□	□	1	0	0	0	#	1	0	1	1	□	□…
…□	□	0	0	0	0	#	#	0	1	1	□	□…
…□	□	0	0	0	0	#	#	#	1	1	□	□…
…□	□	0	0	1	0	#	#	#	#	1	□	□…
…□	□	0	0	1	1	#	#	#	#	#	□	□…
…□	□	0	0	1	1	□	□	□	□	□	□	□…

b) f_T berechnet das bitweise exklusive Oder auf w_1 und w_2:

$$f_T\colon \bigcup_{i=0}^{\infty} \left(\{0,1\}^i \times \{0,1\}^i \right) \to \{0,1\}^*$$

$$f_T(w_1, w_2) = w_3 \in \{0,1\}^{|w_1|}$$

$$\text{wobei für } 0 \leq j < |w_1| : w_3(j) = \begin{cases} 0 & | \; w_1(j) = w_2(j) \\ 1 & | \; w_1(j) \neq w_2(j) \end{cases}$$

c) Zur Bestimmung von $Time_T(n)$ genügt es, Eingabeworte der Form $w = w_1 \# w_2$ mit $|w_1| = |w_2|$ zu betrachten, denn für alle anderen Eingaben bricht die Berechnung nur früher ab als für ein Wort dieser Form mit der gleichen Länge. Es gilt also im Folgenden $n = 2k + 1$, wobei $k = |w_1| = \lfloor n/2 \rfloor$. Bei der Abarbeitung des Eingabewortes w durchläuft T stets die gleiche Folge von Zustandsübergängen:

i) Markieren des linkesten Zeichens in w_1, welches noch nicht abgearbeitet wurde

ii) Überspringen der restlichen Zeichen von w_1 sowie aller #

iii) Ersetzen des nächsten Zeichens (in w_2) durch #

iv) Überspringen aller Zeichen bis zu dem in w_1 markierten

v) Ersetzen des markierten Zeichens

Diese Abfolge wird k Mal durchlaufen. Dabei umfassen (ii) und (iv) k Schritte, da jeweils k Zeichen übersprungen werden (nicht etwa $k + 1$, da ja in den jeweiligen Vorzuständen schon um ein Zeichen nach rechts gerückt wird). (i), (iii) und (v) umfassen jeweils einen Schritt. Wurden alle Zeichen aus dem ursprünglichen Wort w_2 durch # ersetzt, werden in weiteren $k + 1$

Schritten sämtliche # gelöscht. Insgesamt ergibt sich also:

$$\begin{aligned}
Time_T(n) &= k \cdot (2 \cdot (k+1) + 3 \cdot 1) + k + 1 \\
&= \lfloor n/2 \rfloor \cdot (2 \cdot (\lfloor n/2 \rfloor + 1) + 3) + \lfloor n/2 \rfloor + 1 \\
&= 2\lfloor n/2 \rfloor^2 + 2\lfloor n/2 \rfloor + 3\lfloor n/2 \rfloor + \lfloor n/2 \rfloor + 1 \\
&= 2\lfloor n/2 \rfloor^2 + 6\lfloor n/2 \rfloor + 1 \\
&\in \Theta(n^2)
\end{aligned}$$

Da T nur den Bereich auf dem Band verwendet, auf dem initial das Eingabewort steht, gilt zudem $Space_T(n) \in \Theta(n)$.

Aufgabe 3 - Turingmaschine II (4 Punkte)

Auch in dieser Aufgabe sei das Alphabet $A = \{0, 1, \#\}$. Zudem sei die Sprache $L = \{w_1\#w_2 \in A^* \mid w_1, w_2 \in \{0, 1\}^* \text{ und } N_1(w_1) \geq N_1(w_2)\} \subseteq A^*$ gegeben.

a) Beschreiben Sie möglichst präzise die Arbeitsweise eines Turingmaschinenakzeptors, der L akzeptiert. (1 Punkt)

b) Geben Sie einen Turingmaschinenakzeptor T an, so dass T genau die Sprache L akzeptiert. Die Zustandsmenge von T soll dabei höchstens 8 Zustände umfassen. (3 Punkte)

Lösung 3

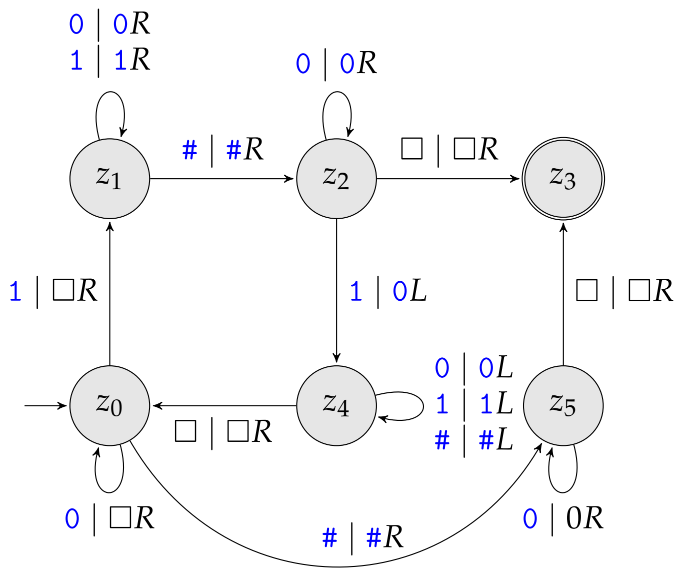
a) Der Turingmaschinenakzeptor löscht sukzessive jeweils eine 1 aus w_1 und anschließend eine 1 aus w_2.

   Dazu wird zunächst vom linkesten Zeichen auf dem Arbeitsband ausgehend die erste 1 vor den ersten Vorkommen von # gesucht und durch □ ersetzt. Ebenso werden alle Vorkommen von 0 ersetzt, die bis dahin gelesen wurden. Anschließend werden alle Zeichen vor einem # übersprungen und die erste 1 nach dem # durch 0 ersetzt. Dabei werden alle Vorkommen von 0 ignoriert. Anschließend läuft der Lesekopf alle Zeichen bis zu der linkesten nicht-leeren Stelle auf dem Arbeitsband und der Prozess wiederholt sich.

   Gibt es nach dem Ersetzen der 1 rechts von # keine weitere 1, wird die Eingabe akzeptiert. Wurden alle Zeichen links von # gelöscht, so geht die Turingmaschine genau dann in einen akzeptierenden Zustand über, wenn rechts von # lediglich 0 bis zum ersten □ gelesen wird.

Kann kann in einem Schritt der Abarbeitung der Eingabe kein Übergang wie eben beschrieben gemacht werden, hält die Turingmaschine und akzeptiert nicht.

b) Der Turingmaschinenakzeptor zur Beschreibung aus Teilaufgabe (a):



Aufgabe 4 - Das aussagenlogische Koinzidenzlemma (4 Punkte)

Sei Var_{AL} eine nicht-leere endliche Menge von aussagenlogischen Variablen und sei $\mathtt{P} \in Var_{AL}$ eine aussagenlogische Variable. Zeigen Sie die folgende Behauptung mittels struktureller Induktion über aussagenlogischen Formeln:

> Für eine beliebige aussagenlogische Formel F, in der $\mathtt{P}$ nicht vorkommt, und Interpretationen I, I' mit $I(\mathtt{P}) \neq I'(\mathtt{P})$ und $I(\mathtt{Q}) = I'(\mathtt{Q})$ für $\mathtt{Q} \in Var_{AL} \setminus \{\mathtt{P}\}$ gilt $val_I(F) = val_{I'}(F)$.

Dieses Theorem nennt man auch das *aussagenlogische Koinzidenzlemma*.

Lösung 4

Wir beweisen die Aussage mit Hilfe von struktureller Induktion über aussagenlogischen Formeln:

IA: Die kürzestmöglichen aussagenlogischen Formeln enthalten eine einzige aussagenlogische Variable und keine Operatoren, also $F = \mathtt{B}$ mit $\mathtt{B} \in Var_{AL}$. Nach Voraussetzung gilt $\mathtt{B} \in Var_{AL} \setminus \{\mathtt{A}\}$, also $val_I(F) = I(\mathtt{B}) = I'(\mathtt{B}) = val_{I'}(F)$ ✓

IS: Sei F eine nichtatomare aussagenlogische Formel.

- Fall 1: $F = (\neg G)$ für $G \in For_{AL}$

  Da F die Variable $\mathtt{P}$ nicht enthält, enthält auch G $\mathtt{P}$ nicht. Per Induktionsannahme gilt zudem $val_I(G) = val_{I'}(G)$. Damit:

$$val_I(F) = \left\{ \begin{array}{ll} \mathbf{w} & \text{falls } val_I(G) = \mathbf{f} = val_{I'}(G) \\ \mathbf{f} & \text{falls } val_I(G) = \mathbf{w} = val_{I'}(G) \end{array} \right\} = val_{I'}(F)$$

- Fall 2: $F = (G_1 \wedge G_2)$ für $G_1, G_2 \in For_{AL}$

  Da F die Variable $\mathtt{P}$ nicht enthält, enthalten weder G_1 noch G_2 $\mathtt{P}$. Per Induktionsannahme gilt zudem $val_I(G_1) = val_{I'}(G_1)$ und $val_I(G_2) = val_{I'}(G_2)$. Damit:

$$\begin{aligned} val_I(F) &= \begin{cases} \mathbf{w} & \text{falls } val_I(G_1) = \mathbf{w} = val_I(G_2) \\ \mathbf{f} & \text{sonst} \end{cases} \\ &= \begin{cases} \mathbf{w} & \text{falls } val_{I'}(G_1) = \mathbf{w} = val_{I'}(G_2) \\ \mathbf{f} & \text{sonst} \end{cases} \\ &= val_{I'}(F) \end{aligned}$$

- Fall 3: $F = (G_1 \vee G_2)$ für $G_1, G_2 \in For_{AL}$

  Da F die Variable $\mathtt{P}$ nicht enthält, enthalten weder G_1 noch G_2 $\mathtt{P}$. Per Induktionsannahme gilt zudem $val_I(G_1) = val_{I'}(G_1)$ und $val_I(G_2) = val_{I'}(G_2)$. Damit:

$$\begin{aligned} val_I(F) &= \begin{cases} \mathbf{w} & \text{falls } val_I(G_1) = \mathbf{w} \text{ oder } val_I(G_2) = \mathbf{w} \\ \mathbf{f} & \text{sonst} \end{cases} \\ &= \begin{cases} \mathbf{w} & \text{falls } val_{I'}(G_1) = \mathbf{w} \text{ oder } val_{I'}(G_2) = \mathbf{w} \\ \mathbf{f} & \text{sonst} \end{cases} \\ &= val_{I'}(F) \end{aligned}$$

- Fall 4: $F = (G_1 \rightarrow G_2)$ für $G_1, G_2 \in For_{AL}$

Da F die Variable `P` nicht enthält, enthalten weder G_1 noch G_2 `P`. Per Induktionsannahme gilt zudem $val_I(G_1) = val_{I'}(G_1)$ und $val_I(G_2) = val_{I'}(G_2)$. Damit:

$$\begin{aligned} val_I(F) &= \begin{cases} \mathbf{f} & \text{falls } val_I(G_1) = \mathbf{w} \text{ und } val_I(G_2) = \mathbf{f} \\ \mathbf{w} & \text{sonst} \end{cases} \\ &= \begin{cases} \mathbf{f} & \text{falls } val_{I'}(G_1) = \mathbf{w} \text{ und } val_{I'}(G_2) = \mathbf{f} \\ \mathbf{w} & \text{sonst} \end{cases} \\ &= val_{I'}(F) \end{aligned}$$

Also gilt in jedem Fall $val_I(F) = val_{I'}(F)$. □

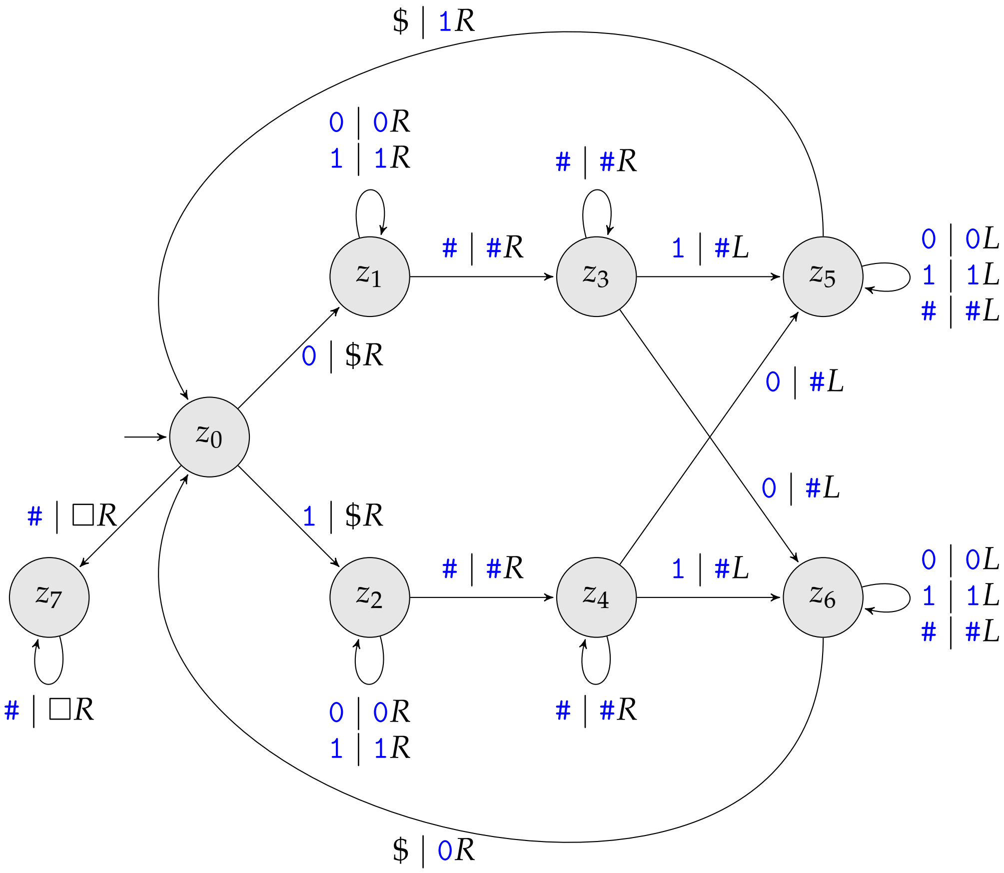


Abbildung 1: Turingmaschine T aus Aufgabe 2

… □	□	1	0	0	0	#	1	0	1	1	□	□ …
… □												□ …
… □												□ …
… □												□ …
… □												□ …
… □												□ …
… □												□ …
… □												□ …
… □												□ …
… □												□ …
… □												□ …

Abbildung 2: Raster für Aufgabe 2(a)

Bitte beachten Sie die folgenden Hinweise:

- Lösungen **müssen** handschriftlich erstellt werden
- Ihre Abgabe sollte die erste Seite dieser Datei als Deckblatt haben
- Ihre Abgabe muss **rechtzeitig** erfolgen

Außerdem, wenn Sie Ihre Ausarbeitung über die Abgabekästen im Keller des Informatik-Gebäudes abgeben:

- Ihre Abgabe muss in der oberen **linken** Ecke zusammengeheftet werden
- Tablet-Ausdrucke sind zulässig

Wenn Sie Ihre Ausarbeitung online über ILIAS abgeben, dann achten Sie darauf:

- Ihre Abgabe muss **genau eine** PDF-Datei sein
- Scans und lesbare Fotos sind zulässig
- Abgabe erfolgt unter „Tutorien" im Ordner **Ihres** Tutoriums