Lehrgebiet Theoretische Informatik

Klausur 1
Formale Systeme, Automaten, Prozesse

17.08.2020

VORNAME: NAME:

MATRIKELNUMMER: STUDIENGANG:

Hinweise:

- Als einziges Hilfsmittel ist ein *unkommentiertes* Wörterbuch für Deutsch-Deutsch bzw. Deutsch-Englisch zugelassen. Dieses wird von der Aufsicht während der Klausur kontrolliert.
- Mit der Unterschrift erklären Sie, dass Sie sich gesundheitlich in der Lage fühlen, diese Klausur mitzuschreiben.
- Jedes Lösungsblatt ist mit Name und Matrikelnummer zu versehen.
- Schreiben Sie Ihre Lösungen zu einer Aufgabe, falls vorhanden, in die dafür vorgesehene Box. Geben Sie den dazugehörigen Rechenweg auf dem zusätzlichen Klausurpapier an.
- Bitte schreiben Sie deutlich. Unleserliches wird nicht korrigiert und nicht gewertet.
- Schmierblätter werden mit abgegeben; streichen Sie diese durch oder machen Sie sie anderweitig als Schmierblätter kenntlich. Es kann nur ein Lösungsversuch pro Aufgabe gewertet werden. Im Zweifel wird das Falsche gewertet.
- Bitte verwenden Sie einen dokumentenechten Stift mit blauer oder schwarzer Tinte und verwenden Sie keinen Tintenkiller, Tipp-Ex oder Ähnliches. Benutzen Sie nur das zur Verfügung gestellte Papier. Entfernen Sie nicht die Heftklammern.
- Sie haben zwei Stunden zur Bearbeitung der Klausur.
- Bewahren Sie Ihre Taschen vor sich (nach Möglichkeit in der Reihe vor Ihnen) auf dem Boden, halten Sie diese während der gesamten Klausur geschlossen. Ein Zugriff (auch auf die geschlossene) Tasche kann als Täuschungsversuch gewertet werden.
- Sie dürfen sich innerhalb der Klausuraufgaben auf Ergebnisse aus der Vorlesung und Ergebnisse aus den Übungsaufgaben beziehen. Dies bedeutet jedoch nicht, dass Sie eine Aufgabe alleine mit “Wurde bereits in der Vorlesung gezeigt” o.Ä. beantworten dürfen.

Ich versichere, die Klausur selbstständig bearbeitet zu haben. Mir ist bekannt, dass die Klausur bei einem Täuschungsversuch mit „nicht bestanden“ bewertet wird.

....
(Unterschrift)

Aufgabe	1	2	3	4	Σ
Punkte	18	20	18	24	80
erreicht					

Note: ....

Formale Systeme, Automaten, Prozesse, SS 2020
Prof. Dr. P. Rossmanith
J. Dreier, T. Hartmann, H. Lotze, D. Mock

Datum: 17.08.2020

Klausur mit Lösungen 01

Aufgabe K1 (4+6+2+6 Punkte)
Ziel dieser Aufgabe ist es, einen einfachen regulären Ausdruck für die Sprache $L = (ab^*cd^*)^* \cap (a^*bc^*d)^*$ zu konstruieren.

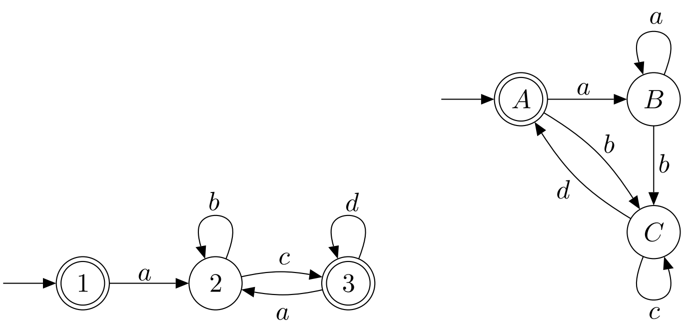
a) Finden Sie einfache NFAs für die beiden Sprachen $(ab^*cd^*)^*$ und $(a^*bc^*d)^*$ (ohne Beweis).

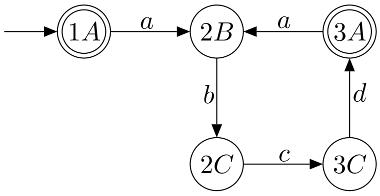
b) Konstruieren Sie den Produktautomaten der beiden, der den Schnitt erkennt.

c) Lesen Sie einen regulären Ausdruck für die Sprache L aus dem Automaten aus b) ab.

d) Es seien jetzt A und B beliebige reguläre Sprachen. Gibt es dann immer eine reguläre Sprache C, so dass $A^* \cap B^* = C^*$ gilt? Beweisen oder widerlegen Sie.

Lösungsvorschlag

a) 

b) 

c) $(abcd)*$

d) Es sei $C = A^* \cap B^*$, welche regulär ist, da reguläre Sprachen unter $\cap$ und * abgeschlossen sind. Wir müssen zeigen, dass $C^* = (A^* \cap B^*)^* = A^* \cap B^*$. Es gilt, dass $(A^* \cap B^*)^* \supseteq A^* \cap B^*$ nach Definition von *. Für die andere Teilmengen-Beziehung, sei $w \in (A^* \cap B^*)^*$. Dann gibt es ein $n \in \mathbf{N}_0$ und x_i so dass $x_1 \dots x_n = w$ und $x_i \in A^* \cap B^*$. Dann ist $x_i \in A^*$ und demnach auch $w \in A^*$. Analog ist $x_i \in B^*$ und $w \in B^*$. Somit ist $w \in A^* \cap B^*$.

Aufgabe K2 (2+3+2+3+2+5+3 Punkte)
Sei G die folgende kontextfreie Grammatik:

$$\begin{array}{lcl} S & \rightarrow & B \mid BA \mid BB \\ A & \rightarrow & SEc \mid SAdE \\ B & \rightarrow & bDB \mid dS \mid DdD \mid DaA \\ C & \rightarrow & aA \mid aFE \\ D & \rightarrow & aDS \mid \varepsilon \mid SBdE \\ E & \rightarrow & AEd \mid AdD \\ F & \rightarrow & b \mid CS \end{array}$$

a) Die unerreichbaren Symbole sind definiert als $\{\, A \in N \mid$ es gibt kein $\alpha A \beta \in (N \cup T)^*$ mit $S \overset{*}{\Rightarrow} \alpha A \beta \,\}$. Welche Symbole sind unerreichbar?

Begründen Sie im Folgendem jeweils Ihre Antwort, zum Beispiel mit einer geeigneten *pre**-Konstruktion. (Eine Antwort wie "ja", "nein" oder "S" reicht nicht aus.)

b) Die unproduktiven Symbole einer Grammatik sind definiert als:

$$\{\, A \in N \mid \text{es gibt kein } w \in T^* \text{ mit } A \overset{*}{\Rightarrow} w \,\}$$

Wie lauten die unproduktiven Symbole von G?

c) Gilt $L(G) = \emptyset$?

d) Gilt $\varepsilon \in L(G)$?

e) Die nullierbaren Symbole einer Sprache sind alle Symbole, die sich nach ε ableiten lassen. Wie lauten die nullierbaren Symbole von G?

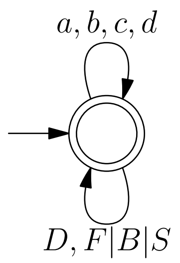
f) Leiten Sie aus G eine kontextfreie Grammatik G' ab, so dass $L(G') = L(G)$ und G' keine unproduktiven, unerreichbaren oder nullierbaren Symbole enthalten.

g) Gilt $badd \in L(G)$?

Lösungsvorschlag
Hinweis: Zwischenschritte bei pre sind nicht verlangt. Der saturierte Automat reicht fuer pre* selbst. Natuerlich mussen Antwortsaetze angegeben werden.*

a) Die Symbole C und F sind unerreichbar. Die Symbole A und B sind direkt von S erreichbar, E und D sind jeweils von A bzw. B erreichbar. Alle Ableitungensregeln von S, A, B, E, D führen wiederum nur zu Non-Terminalen aus S, A, B, E, D. Also sind die verbleibenden C, F unerreichbar.

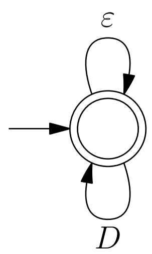
b) Die unproduktiven Symbole sind $N \setminus pre^*(T^*)$. Der Automat für $pre^*(T^*)$ lautet:



Damit ist $pre^*(T^*) = \{S, B, D, F, a, b, c, d\}^*$ und die unproduktiven Symbole sind A, C und E.

c) Da S produktiv ist, kann von S ein $w \in T^*$ abgeleitet werden. Somit ist die von G erzeugte Sprache nicht leer.

d) Es gilt $\varepsilon \in L(G)$ genau dann wenn $S \in pre^*(\{\varepsilon\})$. Der Automat für $pre^*(\{\epsilon\})$ lautet:



Es gilt $pre^*(\{\varepsilon\}) = D^*$. Somit enthält $L(G)$ nicht das leere Wort.

e) Die nullierbaren Symbole der Grammatik sind genau die Nichtterminale in $pre^*(\{\varepsilon\})$. Diese Menge haben wir bereits in Teilaufgabe d) bestimmt, d.h. nur D läßt sich nach ε ableiten.

f) *Hinweis: Die Erklaerung ist nicht notwenig* Regeln mit unerreichbaren oder unproduktiven Symbolen dürfen direkt gelöscht werden, da sie, in einer Ableitung vom Startsymbol aus, entweder nicht verwendet werden können (unerreichbar) oder bei Verwendung eine Ableitung eines Wortes aus Terminalsymbolen unmöglich machen (unproduktiv). Nullierbare Symbole können wie folgt behandelt werden:

1) Sei $A \to \varepsilon \in G$ und $A \neq S$.
2) Lösche $A \to \varepsilon$ aus G.
3) Für jede Regel $B \to \alpha A \beta$ füge die Regel $B \to \alpha\beta$ in G ein.
4) Falls keine Regel der Form $A \to \varepsilon$ mehr existiert fertig, ansonsten gehe zu 1.

Somit hat die resultierende Grammatik keine Regeln der Form $A \to \varepsilon$ für $A \neq S$. Die Regel $S \to \varepsilon$ gehört genau dann zur Grammatik, wenn $\varepsilon \in L(G)$ ist.

Für die Grammatik G ergibt sich also folgende Grammatik G':

$$\begin{array}{lcl} S & \to & B \mid BB \\ B & \to & bDB \mid dS \mid DdD \mid bB \mid dD \mid Dd \mid d \\ D & \to & aDS \mid aS \end{array}$$

g) Ja, denn es gibt zum Beispiel folgende Ableitung (für Grammatik G'):

$$S \vdash B \vdash bDB \vdash bDd \vdash baSd \vdash baBd \vdash badd$$

Aufgabe K3 (8+4+6 Punkte)

Es sei $L = \{ w \in \{a,b\}^* \mid w \neq w^R \}$, die Sprache der Nicht-Palindrome aus den Zeichen a und b.

a) Konstruieren Sie einen Kellerautomaten der mit Endzuständen akzeptiert und der L erkennt. Erläutern Sie die Idee, wie er funktioniert. Zeichnen Sie ihn in den Kasten:

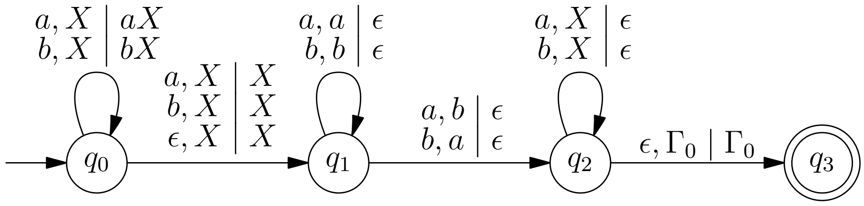
b) Geben Sie eine Konfigurationsfolge Ihres Automaten für jeweils die Wörter $w_1 = abbaba$ und $w_2 = bbbabba$ an.

c) Beweisen oder widerlegen Sie: L ist eine reguläre Sprache.

Lösungsvorschlag

a) Das X in der Grafik ist ein Platzhalter für alle Kellersymbole, also $X \in \{a,b\}$.
Der Automat funktioniert wie folgt: Zuerst wird in q_0 die erste Hälfte des Wortes eingelesen. Bei Worten ungerader Länge werden die ersten $\lfloor \frac{n}{2} \rfloor$ Zeichen eingelesen. In beiden Fällen werden diese auf den Keller geschrieben. Über die Transition noch q_2 rät der Automat, dass die zweite Worthälfte beginnt und ignoriert bei Wörtern ungerader Länge den mittleren Buchstaben.

Nun wird der Rest des Wortes eingelesen und für jeden der eingelesenen Buchstaben wird ein Zeichen vom Keller entfernt. Damit der Lauf einen Endzustand erreichen kann, muss über die Transition von q_1 nach q_2 mindestens ein Buchstabe der zweiten Worthälfte *nicht* einem Zeichen auf dem Stack entsprechen, was eine hinreichende Bedingung für ein nicht-Palindrom darstellt. Anschließend wird der Rest des Eingabewortes gelesen und damit der Rest des Kellers geleert.



b)

Konfigurationsfolge für *abbaba*

$$\begin{aligned}
(abbaba, q_0, \Gamma_0) &\vdash (bbaba, q_0, a\Gamma_0)\\
&\vdash (baba, q_0, ba\Gamma_0)\\
&\vdash (aba, q_0, bba\Gamma_0)\\
&\vdash (aba, q_1, bba\Gamma_0)\\
&\vdash (ba, q_2, ba\Gamma_0)\\
&\vdash (a, q_2, a\Gamma_0)\\
&\vdash (\epsilon, q_2, \Gamma_0)\\
&\vdash (\epsilon, q_3, \Gamma_0)
\end{aligned}$$

Konfigurationsfolge für *bbbabba*

$$\begin{aligned}(bbbabba, q_0, \Gamma_0) &\vdash (bbabba, q_0, b\Gamma_0)\\ &\vdash (babba, q_0, bb\Gamma_0)\\ &\vdash (abba, q_0, bbb\Gamma_0)\\ &\vdash (bba, q_1, bbb\Gamma_0)\\ &\vdash (ba, q_1, bb\Gamma_0)\\ &\vdash (a, q_1, b\Gamma_0)\\ &\vdash (\epsilon, q_2, \Gamma_0)\\ &\vdash (\epsilon, q_3, \Gamma_0)\end{aligned}$$

c) Die Sprache ist nicht regulär. Aus der Vorlesung ist bekannt, dass die Sprache K aller Palindrome nicht regulär ist. Da reguläre Sprachen unter Komplement abgeschlossen sind und $\bar{K} = L$ gilt, kann damit L nicht regulär sein.

Aufgabe K4 (8+8+4+4 Punkte)
Wir definieren einen *Zwei-Keller-Automaten* M wie folgt, welcher fast identisch zu den bekannten Kellerautomaten mit Endzuständen ist. $M = (Q, \Sigma, \Gamma, \delta, q_0, \Gamma_0, F)$, wobei sich nur δ von der Definition der Vorlesung unterscheidet: $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times \Gamma \to 2^{Q \times \Gamma^* \times \Gamma^*}$.
In diesem Automatenmodell gibt es also zwei Keller, von deren Inhalt eine Transition abhängt und auf die bei jeder Transition zugegriffen wird.

a) Beweisen Sie mit Hilfe des Pumping-Lemmas für kontextfreie Sprachen, dass die Sprache $L = \{\, ww \mid w \in \{a, b\}^* \,\}$ nicht kontextfrei ist.

b) Geben Sie graphisch in der Box unten einen Zwei-Keller-Automaten an, der L erkennt. Erläutern Sie ihre Notation und die Funktionsweise Ihres Automaten ausführlich.

c) Geben Sie eine Konfigurationsfolge Ihres Automaten für jeweils die Wörter $w_1 = \varepsilon$ und $w_2 = aabaab$ an.

d) Vergleichen Sie die Menge der Sprachen, die von Zwei-Keller-Automaten erkannt werden können mit der Menge der kontextfreien Sprachen (echte Teilmenge, gleich, echte Obermenge, weder noch). Begründen Sie Ihre Antwort!

Lösungsvorschlag

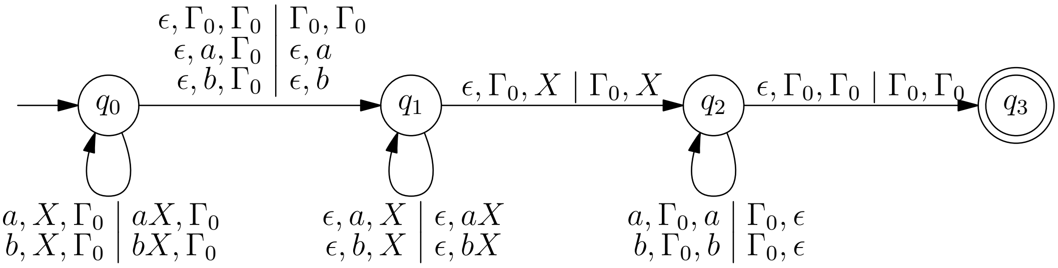
a) Angenommen L sei kontextfrei. Dann existiert ein n, so dass jedes Wort $z \in L$ mit $|z| \geq n$ in $z = uvwxy$ zerlegt werden kann mit

   1. $|vwx| \leq n$,
   2. $|vx| \geq 1$ und
   3. $uv^iwx^iy \in L$ für alle $i \in \mathbf{N}_0$.

   Sei $z = a^nb^na^nb^n$. Offensichtlich gilt $z \in L$. Falls $vwx = a^pa^qa^r$ oder $vwx = b^pb^qb^r$ für $p+q+r \leq n$ und $p+r \geq 1$ ist für $i = 0$ das Wort $uv^0wx^0y = a^qb^na^nb^n, a^nb^qa^nb^n, a^nb^na^qb^n$ oder $a^nb^na^nb^q$, von denen offensichtlich keines in L liegt (da $q < n$). Falls vwx komplett in a^nb^n liegt (und $vx = a^pb^q$) (symmetrisch ob im vorderen oder hinteren Teil), lautet das resultierende Wort für $i = 0$ wie folgt: $uv^0wx^0y = a^{n-p}b^{n-r}a^nb^n$ mit $p + r \geq 1$ und ist damit wiederrum nicht in L. Zuletzt betrachten wir den Fall, dass vwx komplett in b^na^n enthalten ist. Analog lautet das resultierende Wort für $i = 0$ in diesem Fall wie folgt: $uv^0wx^0y = a^nb^{n-p}a^{n-q}b^n$, welches auch nicht in L liegt.

   Aus dem Pumping-Lemma für kontextfreie Sprachen folgt, dass L nicht kontextfrei ist.

b) Wir beschriften Transitionen wie folgt: $a, A, X \mid B, Y$, wobei A das oberste Symbol auf dem linken Stack und X das auf dem rechten ist, welche durch je B bzw. Y ersetzt werden.



   Der Automat erkennt ein Wort $w = w_1w_1 \in L$ wie folgt: Zuerst wird in q_0 das erste Teilwort w_1 eingelesen und auf den ersten Stack geschrieben. Der Automat rät nun nichtdeterministisch, dass w_1 vollständig eingelesen wurde und geht in q_1 über. In q_1 wird das nun (rückwärts) auf dem ersten Stack liegende Wort (vorwärts) auf den zweiten Stack umgeschrieben. Es kann erst in q_2 übergegangen werden, sobald der erste der zwei Stacks vollständig geleert wurde. Anschließend lesen wir den Rest von w ein und löschen die Symbole von dem zweiten Stack, dabei muss jeder Buchstabe der Eingabe auch auf dem Stack vorhanden sein, es kann also nur ein zweites Mal w_1 gelesen werden, ohne dass der Automat verklemmt. Schließlich gehen wir in den Endzustand q_3 über, sobald beide Stacks leer und das Eingabewort komplett gelesen wurde.

c) $(q_0, aabaab, \Gamma_0, \Gamma_0) \vdash (q_0, abaab, a\Gamma_0, \Gamma_0) \vdash (q_0, baab, aa\Gamma_0, \Gamma_0) \vdash (q_1, aab, baa\Gamma_0, \Gamma_0) \vdash (q_1, aab, aa\Gamma_0, b\Gamma_0) \vdash (q_1, aab, a\Gamma_0, ab\Gamma_0) \vdash (q_1, aab, \Gamma_0, aab\Gamma_0) \vdash (q_2, aab, \Gamma_0, aab\Gamma_0) \vdash (q_2, ab, \Gamma_0, ab\Gamma_0) \vdash (q_2, b, \Gamma_0, b\Gamma_0) \vdash (q_2, \varepsilon, \Gamma_0, \Gamma_0) \vdash (q_3, \varepsilon, \Gamma_0, \Gamma_0)$

$(q_0, \varepsilon, \Gamma_0, \Gamma_0) \vdash (q_1, \varepsilon, \Gamma_0, \Gamma_0) \vdash (q_2, \varepsilon, \Gamma_0, \Gamma_0) \vdash (q_3, \varepsilon, \Gamma_0, \Gamma_0)$.

d) Das Zwei-Keller-Modell kann in der Tat mindestens die kontextfreien Sprachen erkennen: Aus der Vorlesung ist bekannt, dass Kellerautomaten genau die kontextfreien Sprachen erkennen. Wir können mit Hilfe des Zwei-Keller-Modells jeden Kellerautomaten simulieren, indem wir während des Laufes nicht auf den zweiten Keller zugreifen.

Aus der vorherigen Aufgabenteilen folgt, dass das Zwe-Keller-Modell auch einige (nicht unbedingt alle!) nicht-kontextfreie Sprachen erkennt. Damit ist Menge der durch Zwei-Keller-Automaten erkannnten Sprachen eine echte Obermenge der der kontextfreien Sprachen.

Formale Systeme, Automaten, Prozesse, SS 2020
Prof. Dr. P. Rossmanith
J. Dreier, T. Hartmann, H. Lotze, D. Mock

RWTH AACHEN UNIVERSITY

Datum: 17.08.2020

Klausur mit Lösungen 01

Formale Systeme, Automaten, Prozesse, SS 2020
Prof. Dr. P. Rossmanith
J. Dreier, T. Hartmann, H. Lotze, D. Mock

Datum: 17.08.2020

Klausur mit Lösungen 01

Formale Systeme, Automaten, Prozesse, SS 2020
Prof. Dr. P. Rossmanith
J. Dreier, T. Hartmann, H. Lotze, D. Mock

Datum: 17.08.2020

Klausur mit Lösungen 01

Formale Systeme, Automaten, Prozesse, SS 2020
Prof. Dr. P. Rossmanith
J. Dreier, T. Hartmann, H. Lotze, D. Mock

Datum: 17.08.2020

Klausur mit Lösungen 01

Formale Systeme, Automaten, Prozesse, SS 2020
Prof. Dr. P. Rossmanith
J. Dreier, T. Hartmann, H. Lotze, D. Mock

Datum: 17.08.2020

Klausur mit Lösungen 01

Formale Systeme, Automaten, Prozesse, SS 2020
Prof. Dr. P. Rossmanith
J. Dreier, T. Hartmann, H. Lotze, D. Mock

Datum: 17.08.2020

Klausur mit Lösungen 01