Übungsblatt 10 - mit Lösungen

Probeklausur

Maximal erreichbar: **100 Punkte**

Theoretische Informatik
Angewandte Informatik
Wintersemester 2015/2016
Prof. Barbara Staehle, HTWG Konstanz

Hinweise:

- Falls Sie für die Aufgaben alle Punkte haben wollen, begründen Sie Ihre Antworten, bzw. stellen Sie den Lösungs- / Rechenweg nachvollziehbar dar.
- Sie müssen weder zum Bestehen noch für eine sehr gute Note alle Aufgaben korrekt bearbeiten. Zum Bestehen reichen ca. 40-50 Punkte, eine sehr gute Note gibt es ab ca. 85 Punkten.

Aufgabe 10.1 Sprachen, Grammatiken und Automaten

Gegeben sei das Alphabet $\Sigma = \{0, 1, 2, \ldots, 9\}$.

Teilaufgabe 10.1.1 6 Punkte

Geben Sie die folgenden Sprachen über Σ an **oder** beschreiben Sie diese mit Hilfe von Beispielen:

1. Σ^0
2. Σ^1
3. Σ^3
4. Σ^*

Lösung

1. $\Sigma^0 = \{\varepsilon\}$, alle Wörter der Länge 0
2. $\Sigma^1 = \Sigma = \{0, 1, 2, \ldots, 9\}$, alle Wörter der Länge 1
3. $\Sigma^3 = \{000, 001, 002, \ldots, 999\}$, alle Wörter der Länge 3
4. $\Sigma^* = \{\varepsilon, 0, 1, 2, \ldots, 123, \ldots, 562864, \ldots\}$, alle Wörter aller Längen

Teilaufgabe 10.1.2 10 Punkte

Wir betrachten die Menge der durch 5 teilbaren natürlichen Zahlen als formale Sprache über Σ :

$$L_5 = \{5, 10, 15, 20 \ldots\} \subseteq \Sigma^*.$$

Geben Sie die Grammatik G_5 an, welche L_5 erzeugt.

Hinweise:

- Machen Sie sich vorab klar, wie man eine durch 5 teilbare Zahl erkennt.
- Form und Menge Ihrer Regeln haben keinen Einfluss auf die Punktgebung, nur deren Funktionalität.

Lösung

Vorüberlegungen:

- Eine durch 5 teilbare Zahl endet immer mit 0 oder 5.
- Die 0 ist nicht durch 5 teilbar (siehe Angabe, $0 \notin \mathbb{N}$).
- Mehrstellige Zahlen beginnen nicht mit führender 0.

Dies führt zur Lösung $G_5 = (N, \Sigma, P, S)$ mit $\mathscr{L}(G_5) = L_5$:

- $N = \{S, S_2\}$ bzw. alternativ $N' = \{S, S_2, S_3\}$
- $\Sigma = \{0, 1, \ldots 9\}$
- P gegeben durch: $\begin{array}{lcl} S & \rightarrow & 5 \mid 1S_2 \mid 2S_2 \mid \ldots \mid 9S_2 \\ S_2 & \rightarrow & 0 \mid 5 \mid 0S_2 \mid 1S_2 \mid \ldots \mid 9S_2 \end{array}$
- Alternative Möglichkeit P': $\begin{array}{lcl} S & \rightarrow & 5 \mid S_2 0 \mid S_2 5 \\ S_2 & \rightarrow & 1S_3 \mid 2S_3 \mid \ldots \mid 9S_3 \\ S_3 & \rightarrow & 0S_3 \mid 1S_3 \mid \ldots \mid 9S_3 \end{array}$

Teilaufgabe 10.1.3 10 Punkte

Geben Sie den DEA A_5 an, welcher L_5 akzeptiert. Geben Sie δ in tabellarischer Form **oder** durch ein Zustandsübergangsdiagramm an.

Hinweise:

- Machen Sie sich vorab klar, wie man eine durch 5 teilbare Zahl erkennt bzw. nutzen Sie Ihre Überlegungen aus Aufgabe 10.1.2.
- Form und Größe Ihres Automaten haben keinen Einfluss auf die Punktgebung, nur dessen Funktionalität.

Lösung

Vorüberlegungen: Siehe Aufgabe 10.1.2. Achte insbesondere auf korrekte Behandlung von Zahlen mit führender 0.

Konstruiere also A_5 mit $\mathscr{L}(A_5) = L_5$ als $A_5 = (S, \Sigma, \delta, F, s_0)$ mit

- $S = \{s_0, s_1, s_2\}$
- $\Sigma = \{0, 1, \ldots 9\}$
- $F = \{s_1\}$
- δ gegeben durch Abbildung 1. Alternative Lösungen, vor allem ohne Fehlerzustand s_3 natürlich möglich.

Teilaufgabe 10.1.4 3 Punkte

1. Welchen Chomsky-Typ hat Ihre Grammatik G_5 aus Aufgabe 10.1.2?
2. Was sagt Ihnen das Ergebnis von Aufgabe 10.1.3 über den maximal möglichen (numerisch größten) Chomsky-Typ einer die Sprache L_5 erzeugenden Grammatik?

Begründen Sie Ihre Antworten.

δ	0	5	$\Sigma \setminus \{0,5\}$
s_0	s_3	s_1	s_2
s_1	s_1	s_1	s_2
s_2	s_1	s_1	s_2
s_3	s_3	s_3	s_3

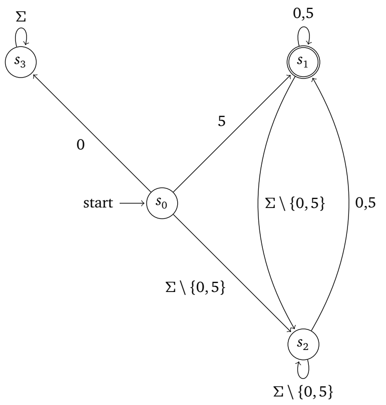

Abbildung 1: Zustandsübergangstabelle und -diagramm für A_5

Lösung

- G_5 mit Regeln P ist regulär, da links nur ein Nonterminal, rechts nur Terminal bzw. Terminal Nonterminal, mit Regeln P' kontextfrei, da links nur ein Nonterminal.
- L_5 wird von einem DEA akzeptiert, damit ist sie regulär, es muss also auch eine reguläre Grammatik (Typ 3) existieren, welche L_5 erzeugt.

Aufgabe 10.2 Sprachen und Automaten

Wir betrachten das Alphabet $\Sigma = \{a, b\}$ und die Sprache $L \subseteq \Sigma^*$.

Wichtig: Die Sprache L wird durch Aufgabe 10.2.1 definiert und in den folgenden Teilaufgaben weiter verwendet.

- Falls Sie Aufgabe 10.2.1 falsch lösen, können Sie trotzdem alle Punkte auf die folgenden Aufgaben bekommen (Folgefehler).
- Falls Sie Aufgabe 10.2.1 **nicht** lösen können, so melden Sie sich bitte, um die Lösung der Aufgabe (die Sprache L) zu erhalten. In diesem Fall werden Ihnen auf Aufgabe 10.2.1 0 Punkte angerechnet, auf die Punkte der weiteren Teilaufgaben hat dies aber keinen negativen Einfluss.

Teilaufgabe 10.2.1 4 Punkte

L wird vom regulären Ausdruck r erzeugt: $\mathscr{L}(r) = L$ mit

$$\begin{aligned} r &= (a|b)^* ab && \text{(formale Schreibweise)} \\ r &= [ab]^* ab && \text{(UNIX-Schreibweise)} \end{aligned}$$

Geben Sie L (z.B. in beschreibender Form), sowie 5 beispielhafte Wörter der Sprache L an.

Lösung

- L enthält alle Wörter, die auf ab enden: $L = \{\omega \in \Sigma^* \mid \omega = \nu ab,\ \nu \in \Sigma^*\} = \{\omega \in \Sigma^* \mid \omega \text{ endet auf } ab\}$
- Beispielhafte Wörter aus L: $ab, aab, baab, abab, aaabbbbabbab$

Teilaufgabe 10.2.2 10 Punkte

Geben Sie den NEA N_L an, welcher L akzeptiert. Geben Sie δ in tabellarischer Form **und** durch ein Zustandsübergangsdiagramm an. Achten Sie darauf, dass Ihr Automat die Vorteile des Nichtdeterminismus auch tatsächlich ausschöpft.

Lösung

Konstruiere N_L mit $\mathscr{L}(N_L) = L$ als $N_L = (S, \Sigma, \delta, F, s_0)$ mit

- $S = \{s_0, s_1, s_2\}$
- $\Sigma = \{a, b\}$
- $F = \{s_2\}$
- δ gegeben durch Abbildung 2. Alternative Lösungen möglich, allerdings muss der Automat nichtdeterministische Elemente beinhalten.

δ	a	b
s_0	$\{s_0, s_1\}$	$\{s_0\}$
s_1	$\emptyset$	$\{s_2\}$
s_2	$\emptyset$	$\emptyset$

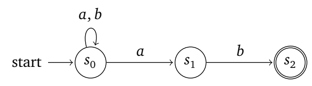

Abbildung 2: Zustandsübergangstabelle und -diagramm für N_L

Teilaufgabe 10.2.3 10 Punkte

Geben Sie den DEA A_L an, welcher L akzeptiert. Geben Sie δ in tabellarischer Form **und** durch ein Zustandsübergangsdiagramm an.

Lösung

Konstruiere A_L mit $\mathscr{L}(N_L) = L$ als $A_L = (S, \Sigma, \delta, F, s_0)$ mit

- $S = \{s_0, s_1, s_2\}$
- $\Sigma = \{a, b\}$
- $F = \{s_2\}$
- δ gegeben durch Abbildung 3. Alternative Lösungen möglich.

δ	a	b
s_0	s_1	s_0
s_1	s_1	s_2
s_2	s_1	s_0

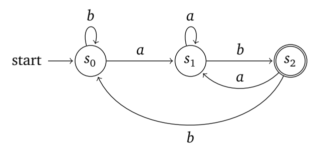

Abbildung 3: Zustandsübergangstabelle und -diagramm für A_L

Aufgabe 10.3 Berechenbarkeit

Gegeben ist das Programm *mystery.while* (siehe Tabelle 1):

```
x1 := succ(x1);
WHILE x1 DO
        x0 := succ(x0);
        LOOP x2 DO
                x1 := pred(x1)
        END
END
x0 := pred(x0);
```

Tabelle 1: mystery.while

Teilaufgabe 10.3.1 9 Punkte

Geben Sie

1. die Änderungen des Eingabevektors v bis das Programm beendet ist (vereinfachten Notation),
2. das Ergebnis der Berechnung des Programms,

in den folgenden Fällen an:

1. $v = (0,4,2)$
2. $v = (0,5,2)$
3. $v = (0,5,0)$

Hinweise:

- Geben Sie alle Zustandsänderungen an, auch wenn sich der Nachfolgezustand nicht vom Vorgängerzustand unterscheidet.
- Falls Sie den Eindruck haben, das Programm läuft in eine unendlich Schleife, dann brechen Sie Ihre Berechnung bitte ab und vermerken dies entsprechend.

Lösung

1. $(0,4,2) \to (0,5,2) \to (1,5,2) \to (1,4,2) \to (1,3,2) \to (2,3,2) \to (2,2,2) \to (2,1,2) \to (3,1,2) \to (3,0,2) \to (3,0,2) \to (2,0,2)$
   Ergebnis: x0 = 2
2. $(0,5,2) \to (0,6,2) \to (1,6,2) \to (1,5,2) \to (1,4,2) \to (2,4,2) \to (2,3,2) \to (2,2,2) \to (3,2,2) \to (3,1,2) \to (3,0,2) \to (2,0,2)$
   Ergebnis: x0 = 2
3. $(0,5,0) \to (0,6,0) \to (1,6,0) \to (2,6,0) \to (3,6,0) \to (4,6,0) \to \ldots$ da x2 = 0 wird die Loop-Schleife nie betreten, x1 wird also nie dekrementiert und folglich auch nie 0, also wird die While-Schleife unendlich oft durchlaufen.
   Ergebnis: undefiniert

Teilaufgabe 10.3.2 8 Punkte

1. Welche Funktion f_m berechnet *mystery.while* für die Zahlen $x1, x2 \in \mathbb{N}_0$?
2. Handelt es sich bei f_m um eine totale oder eine partielle Funktion?
3. Nennen Sie ein Berechnungsmodell mit welchem f_m **ebenfalls** berechnet werden kann.
4. Nennen Sie ein Berechnungsmodell mit welchem f_m **nicht** berechnet werden kann.

Begründen Sie Ihre Antworten.

Lösung

- *mystery.while* berechnet die ganzzahlige Division von $x1$ durch $x2$, falls $x2 \neq 0$. Andernfalls ist das Ergebnis undefiniert:
$$f_m = \begin{cases} \lfloor \frac{x1}{x2} \rfloor & \text{falls } x2 \neq 0, \\ \bot & \text{falls } x2 = 0. \end{cases}$$
- Aus der Definition von f_m sieht man, dass es sich bei f_m um eine partielle Funktion handelt.
- f_m könnte mit einem Goto-Programm, einer μ-rekursiven Funktion, einer Turing-Maschine oder einer RAM berechnet werden.
- Da f_m partiell ist, kann sie nicht mit einem Loop-Programm oder einer primitiv rekursiven Funktion berechnet werden.

Aufgabe 10.4 Arbeitsweise von Turing-Maschinen

Gegeben Sei die Turing-Maschine $T_x = (S, \Sigma, \Pi, \delta, s_0, \square, F)$ mit

- $S = \{s_0, s_1, s_a, s_b, s_f\}$
- $\Sigma = \{a, b\}$
- $\Pi = \{a, b, \square\}$
- $F = \{s_f\}$
- δ gegeben durch Tabelle 2

δ	a	b	$\square$
s_0	$(s_0, a, \rightarrow)$	$(s_0, b, \rightarrow)$	$(s_1, \square, \leftarrow)$
s_1	$(s_a, \square, \leftarrow)$	$(s_b, \square, \leftarrow)$	$(s_f, \square, \circlearrowleft)$
s_a	$(s_a, a, \leftarrow)$	$(s_b, a, \leftarrow)$	$(s_f, a, \circlearrowleft)$
s_b	$(s_a, b, \leftarrow)$	$(s_b, b, \leftarrow)$	$(s_f, b, \circlearrowleft)$
s_f	-	-	-

Tabelle 2: Zustandsübergangsfunktion von T_x

Teilaufgabe 10.4.1 6 Punkte

Stellen Sie δ mit Hilfe eines Zustandsübergangsdiagramms dar.
Hinweis: Eine Anordnung der Zustände ähnlich zu Abbildung 4 ist am übersichtlichsten.

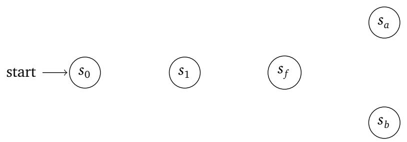

Abbildung 4: Empfohlene Zustandsanordnung für das Zustandsübergangsdiagramm von T_x

LÖSUNG

Siehe Abbildung 5.

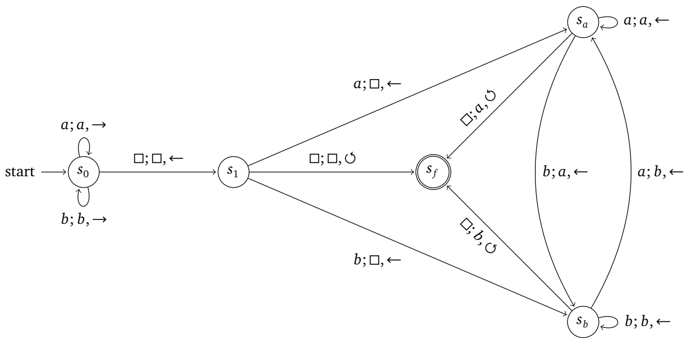

Abbildung 5: Zustandsübergangsdiagramm für T_x

Bedeutung der Zustände (nicht verlangt):

- s_0: Gehe auf äußerst rechtes Zeichen.
- s_1: Äußerstes rechtes Zeichen merken, löschen und nach links gehen.
- s_a: Gemerktes a schreiben, aktuelles Zeichen merken und nach links gehen.
- s_b: Gemerktes b schreiben, aktuelles Zeichen merken und nach links gehen.
- s_f: Endzustand.

TEILAUFGABE 10.4.2 6 PUNKTE

Berechnen Sie die Endkonfiguration von T_x unter der Eingabe von

1. $\omega = bb$
2. $\omega = aba$.

Geben Sie alle Konfigurationen an, welche, ausgehend von der Startkonfiguration, durchlaufen werden.

LÖSUNG

1. $(\square, s_0, bb) \vdash (\square b, s_0, b) \vdash (\square bb, s_0, \square) \vdash (\square b, s_1, b\square) \vdash (\square, s_b, b\square\square) \vdash (\square, s_b, \square b\square\square) \vdash (\square, s_f, bb\square\square)$
2. $(\square, s_0, aba) \vdash (\square a, s_0, ba) \vdash (\square ab, s_0, a) \vdash (\square aba, s_0, \square) \vdash (\square ab, s_1, a\square) \vdash (\square a, s_a, b\square\square) \vdash (\square, s_b, aa\square\square) \vdash (\square, s_a, \square ba\square) \vdash (\square, s_f, aba\square\square)$

TEILAUFGABE 10.4.3 4 PUNKTE

1. Welche Funktion f_x berechnet T_x für ein Eingabewort $\omega \in \Sigma^*$?
2. Handelt es sich bei f_x um eine totale oder eine partielle Funktion?

Begründen Sie Ihre Antworten.

LÖSUNG

1. T_x verschiebt einfach alle Buchstaben des Eingabewortes $\omega \in \Sigma^*$ um eine Stelle nach links. Konkretere Begründung siehe Erklärung der Zustände in der Lösung zu Aufgabe 10.4.1.
2. Da T_x für alle Eingaben erfolgreich terminiert, handelt es sich bei $f_x y$ um eine totale Funktion.

AUFGABE 10.5 WAHR ODER FALSCH?, 14 PUNKTE

Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind. Begründen Sie Ihre Entscheidung (kurz).

1. Für eine atomare Aussage f ist die logische Aussage $f \vee \neg f$ allgemeingültig.
2. Ein Alphabet ist eine endliche oder unendliche Menge von Symbolen.
3. Das Pumpinglemma kann genutzt werden, um zu beweisen, dass eine Sprache regulär ist.
4. Jeder deterministische endliche Automat ist auch ein nichtdeterministischer endlicher Automat.
5. Für einen DEA $A = (S, \Sigma, \delta, F, s_0)$ ist die Relation $R \subseteq \Sigma^* \times \Sigma^* = \{(x, y) \mid x, y \in \Sigma^* \text{ werden von } A \text{ akzeptiert}\}$ eine Äquivalenzrelation.
6. Alle Funktionen die von einer Turing-Maschine berechnet werden können, können auch durch ein Loop-Programm berechnet werden.
7. Eine universelle Turing-Maschine beantwortet die Frage nach dem Leben, dem Universum und dem ganzen Rest.

LÖSUNG

1. Wahr. Für $f = 0$ und $f = 1$ ist die zusammengesetzte Aussage $f \vee \neg f$ immer wahr, das ist gerade die Definition von „allgemeingültig".
2. Falsch. Ein Alphabet ist immer endlich.
3. Falsch. Das Pumpinglemma kann genutzt werden, um zu beweisen, dass eine Sprache **nicht** regulär ist.
4. Wahr. Nur die Umkehrung gilt nicht.
5. Wahr. R ist reflexiv ($(x, x) \in R$), symmetrisch ($(x, y) \in R \Rightarrow (y, x) \in R$) und transitiv ($(x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$)
6. Falsch. Turing-Maschinen können z.B. auch partielle Funktionen berechnen, das können Loop-Programme nicht.
7. Falsch. Eine universelle Turing-Maschine kann die Arbeitsweise aller existierenden Turing-Maschinen nachvollziehen, indem sie als Eingabe die Codierung einer beliebigen Turing-Maschine, sowie deren Eingabe bekommt.