Informatik III - Tutorium XIII & XV (SR -107)

Tut Nr. 4 – Üb3, Nerode Relation, det. Turingmaschine

David Münch

Universität Karlsruhe (TH)
Institut für Informatik
ITI Wagner

21. November 2007

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Organisatorisches

Email: muenchdavid@gmail.com

https://www.stud.uni-karlsruhe.de/~uhbro/

Tutorium 13: Mittwochs 8:00 Uhr - Raum -107
Tutorium 15: Mittwochs 9:45 Uhr - Raum -107

Übungsblattabgabe Donnerstag.

Was wollen wir heute erreichen?

Was wollen wir heute erreichen?

- Wiederholung: Pumping-Lemma korrekt verstehen.

Was wollen wir heute erreichen?

- Wiederholung: Pumping-Lemma korrekt verstehen.
- Mit der Nerode Relation umgehen können.

Was wollen wir heute erreichen?

- Wiederholung: Pumping-Lemma korrekt verstehen.
- Mit der Nerode Relation umgehen können.
- Kapitel **Endliche Sprachen** abschliessen.

Was wollen wir heute erreichen?

- Wiederholung: Pumping-Lemma korrekt verstehen.
- Mit der Nerode Relation umgehen können.
- Kapitel **Endliche Sprachen** abschliessen.
- Einführung in Turingmaschinen.

Probleme

Ab sofort keine unformalen Automaten mehr. Ein Bild ist kein Automat, sondern stellt δ graphisch dar.

Probleme

Ab sofort keine unformalen Automaten mehr. Ein Bild ist kein Automat, sondern stellt δ graphisch dar.

Partnerabgabe ist erwünscht.

Aufgabe 5

Zeigen Sie unter Verwendung des Pumping-Lemmas für folgende Sprachen, dass sie nicht regulär sind. (Hinweis: $|w|_0$ (bzw. $|w|_1$) bezeichnet die Anzahl der Nullen (bzw. Einsen) in w, w^R bezeichnet das Spiegelwort zu w).

Aufgabe 5

Zeigen Sie unter Verwendung des Pumping-Lemmas für folgende Sprachen, dass sie nicht regulär sind. (Hinweis: $|w|_0$ (bzw. $|w|_1$) bezeichnet die Anzahl der Nullen (bzw. Einsen) in w, w^R bezeichnet das Spiegelwort zu w).

a) $L_1 = \{w \in \{a, b\}^* \mid w = w^R\}$

Aufgabe 5

Zeigen Sie unter Verwendung des Pumping-Lemmas für folgende Sprachen, dass sie nicht regulär sind. (Hinweis: $|w|_0$ (bzw. $|w|_1$) bezeichnet die Anzahl der Nullen (bzw. Einsen) in w, w^R bezeichnet das Spiegelwort zu w).

a) $L_1 = \{w \in \{a,b\}^* \mid w = w^R\}$
b) $L_2 = \{a^i b^j c^k \mid i < j < k\}$

Aufgabe 5

Zeigen Sie unter Verwendung des Pumping-Lemmas für folgende Sprachen, dass sie nicht regulär sind. (Hinweis: $|w|_0$ (bzw. $|w|_1$) bezeichnet die Anzahl der Nullen (bzw. Einsen) in w, w^R bezeichnet das Spiegelwort zu w).

a) $L_1 = \{w \in \{a,b\}^* \mid w = w^R\}$
b) $L_2 = \{a^i b^j c^k \mid i < j < k\}$
c) $L_3 = \{w \in \{0,1\}^* \mid \ |w|_0 > |w|_1\}$

Wiederholung

Satz: Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass **für jedes Wort** $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \epsilon,$$

existiert, bei der auch $uv^i x \in L$ ist für alle $i \in \mathbb{N}_0$.

Wiederholung

Satz: Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass **für jedes Wort** $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \epsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Hinweis:
Man verwendet das Pumping-Lemma um zu zeigen, dass eine Sprache nicht regulär ist.

Aufgabe 5a

$$L_1 = \{w \in \{a,b\}^* \mid w = w^R\}$$

Aufgabe 5a

$$L_1 = \{w \in \{a,b\}^* \mid w = w^R\}$$

- Annahme: L_1 ist regulär.

Aufgabe 5a

$$L_1 = \{w \in \{a,b\}^* \mid w = w^R\}$$

- Annahme: L_1 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.

Aufgabe 5a

$$L_1 = \{w \in \{a,b\}^* \mid w = w^R\}$$

- Annahme: L_1 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = a^n b a^n \in L_1$.

Aufgabe 5a

$$L_1 = \{w \in \{a,b\}^* \mid w = w^R\}$$

- Annahme: L_1 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = a^n b a^n \in L_1$.
- Es sei dann $w = uvx$ eine Zerlegung wie im PL mit $|uv| \leq n, |v| \geq 1$.

Aufgabe 5a

$$L_1 = \{w \in \{a, b\}^* \mid w = w^R\}$$

- Annahme: L_1 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = a^n b a^n \in L_1$.
- Es sei dann $w = uvx$ eine Zerlegung wie im PL mit $|uv| \leq n, |v| \geq 1$.
- Aufpumpen. Wähle $i = 2$: $uv^2w = a^{n+|v|}ba^n \notin L_1$.

Aufgabe 5a

$$L_1 = \{w \in \{a,b\}^* \mid w = w^R\}$$

- Annahme: L_1 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = a^n b a^n \in L_1$.
- Es sei dann $w = uvx$ eine Zerlegung wie im PL mit $|uv| \leq n, |v| \geq 1$.
- Aufpumpen. Wähle $i = 2$: $uv^2w = a^{n+|v|}ba^n \notin L_1$.
- Dies ist ein Widerspruch zum PL $\Rightarrow L_1$ ist nicht regulär.

Aufgabe 5b

$$L_2 = \{a^i b^j c^k \mid i < j < k\}$$

Aufgabe 5b

$$L_2 = \{a^i b^j c^k \mid i < j < k\}$$

- Annahme: L_2 ist regulär.

Aufgabe 5b

$$L_2 = \{a^i b^j c^k \mid i < j < k\}$$

- Annahme: L_2 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.

Aufgabe 5b

$$L_2 = \{a^i b^j c^k \mid i < j < k\}$$

- Annahme: L_2 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = a^n b^{n+1} c^{n+2} \in L_2$.

Aufgabe 5b

$$L_2 = \{a^i b^j c^k \mid i < j < k\}$$

- Annahme: L_2 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = a^n b^{n+1} c^{n+2} \in L_2$.
- Es sei dann $w = uvx$ eine Zerlegung wie im PL mit $|uv| \leq n, |v| \geq 1$.

Aufgabe 5b

$$L_2 = \{a^i b^j c^k \mid i < j < k\}$$

- Annahme: L_2 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = a^n b^{n+1} c^{n+2} \in L_2$.
- Es sei dann $w = uvx$ eine Zerlegung wie im PL mit $|uv| \leq n, |v| \geq 1$.
- Aufpumpen. Wähle $i = 2$: $uv^2w = a^{n+|v|} b^{n+1} c^{n+2} \notin L_2$.

Aufgabe 5b

$$L_2 = \{a^i b^j c^k \mid i < j < k\}$$

- Annahme: L_2 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = a^n b^{n+1} c^{n+2} \in L_2$.
- Es sei dann $w = uvx$ eine Zerlegung wie im PL mit $|uv| \leq n, |v| \geq 1$.
- Aufpumpen. Wähle $i = 2$: $uv^2w = a^{n+|v|} b^{n+1} c^{n+2} \notin L_2$.
- Dies ist ein Widerspruch zum PL $\Rightarrow L_2$ ist nicht regulär.

Aufgabe 5c

$$L_3 = \{w \in \{0,1\}^* \mid \ |w|_0 > |w|_1\}$$

Aufgabe 5c

$$L_3 = \{w \in \{0,1\}^* \mid \ |w|_0 > |w|_1\}$$

- Annahme: L_3 ist regulär.

Aufgabe 5c

$$L_3 = \{w \in \{0,1\}^* \mid \ |w|_0 > |w|_1\}$$

- Annahme: L_3 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.

Aufgabe 5c

$$L_3 = \{w \in \{0,1\}^* \mid \ |w|_0 > |w|_1\}$$

- Annahme: L_3 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = 1^n 0^{n+1} \in L_3$.

Aufgabe 5c

$$L_3 = \{w \in \{0,1\}^* \mid \; |w|_0 > |w|_1\}$$

- Annahme: L_3 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = 1^n 0^{n+1} \in L_3$.
- Es sei dann $w = uvx$ eine Zerlegung wie im PL mit $|uv| \leq n, |v| \geq 1$.

Aufgabe 5c

$$L_3 = \{w \in \{0,1\}^* \mid \ |w|_0 > |w|_1\}$$

- Annahme: L_3 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = 1^n 0^{n+1} \in L_3$.
- Es sei dann $w = uvx$ eine Zerlegung wie im PL mit $|uv| \leq n, |v| \geq 1$.
- Aufpumpen. Wähle $i = 2$: $uv^2w = 1^{n+|v|}0^{n+1} \notin L_3$.

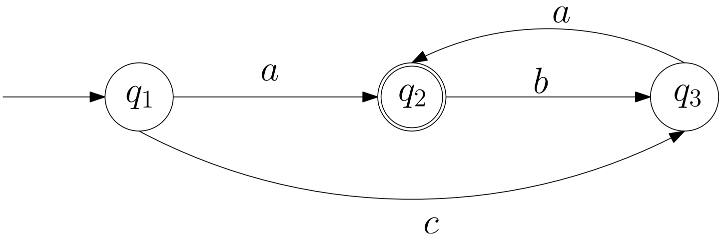
Aufgabe 5c

$$L_3 = \{w \in \{0,1\}^* \mid \; |w|_0 > |w|_1\}$$

- Annahme: L_3 ist regulär.
- Sei $n \in \mathbb{N}$ die Pumping-Lemma Zahl.
- Betrachte $w = 1^n 0^{n+1} \in L_3$.
- Es sei dann $w = uvx$ eine Zerlegung wie im PL mit $|uv| \leq n, |v| \geq 1$.
- Aufpumpen. Wähle $i = 2$: $uv^2w = 1^{n+|v|}0^{n+1} \notin L_3$.
- Dies ist ein Widerspruch zum PL $\Rightarrow L_3$ ist nicht regulär.

Aufgabe 6

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten erkannte Sprache.



Im Folgenden stehe $L_{r,i,t}$ abkürzend für L_{q_r,i,q_t}.

$$
\begin{aligned}
L_{1,0,1} &= \varepsilon & L_{1,1,1} &= \varepsilon \cup \varepsilon\varepsilon^*\varepsilon = \varepsilon & L_{1,2,1} &= \varepsilon \cup a\varepsilon^*\emptyset = \varepsilon \\
L_{1,0,2} &= a & L_{1,1,2} &= a \cup \varepsilon\varepsilon^* a = a & L_{1,2,2} &= a \cup a\varepsilon^*\varepsilon = a \\
L_{1,0,3} &= c & L_{1,1,3} &= c \cup \varepsilon\varepsilon^* c = c & L_{1,2,3} &= c \cup a\varepsilon^* b = c \cup ab \\
L_{2,0,1} &= \emptyset & L_{2,1,1} &= \emptyset \cup \emptyset\varepsilon^*\varepsilon = \emptyset & L_{2,2,1} &= \emptyset \cup \varepsilon\varepsilon^*\emptyset = \emptyset \\
L_{2,0,2} &= \varepsilon & L_{2,1,2} &= \varepsilon \cup \emptyset\varepsilon^* a = \varepsilon & L_{2,2,2} &= \varepsilon \cup \varepsilon\varepsilon^*\varepsilon = \varepsilon \\
L_{2,0,3} &= b & L_{2,1,3} &= b \cup \emptyset\varepsilon^* c = b & L_{2,2,3} &= b \cup \varepsilon\varepsilon^* b = b \\
L_{3,0,1} &= \emptyset & L_{3,1,1} &= \emptyset \cup \emptyset\varepsilon^*\varepsilon = \emptyset & L_{3,2,1} &= \emptyset \cup a\varepsilon^*\emptyset = \emptyset \\
L_{3,0,2} &= a & L_{3,1,2} &= a \cup \emptyset\varepsilon^* a = a & L_{3,2,2} &= a \cup a\varepsilon^*\varepsilon = a \\
L_{3,0,3} &= \varepsilon & L_{3,1,3} &= \varepsilon \cup \emptyset\varepsilon^* c = \varepsilon & L_{3,2,3} &= \varepsilon \cup a\varepsilon^* b = \varepsilon \cup ab
\end{aligned}
$$

$$
\begin{aligned}
L_{1,3,1} &= \varepsilon \cup (c \cup ab)(\varepsilon \cup ab)^*\emptyset = \varepsilon \\
L_{1,3,2} &= a \cup (c \cup ab)(\varepsilon \cup ab)^* a = \mathbf{a \cup (c \cup ab)(ab)^* a} \\
L_{1,3,3} &= c \cup ab \cup (c \cup ab)(\varepsilon \cup ab)^*(\varepsilon \cup ab) = (c \cup ab)(ab)^* \\
L_{2,3,1} &= \emptyset \cup b(\varepsilon \cup ab)^*\emptyset = \emptyset \\
L_{2,3,2} &= \varepsilon \cup b(\varepsilon \cup ab)^* a = \varepsilon \cup (ba)^+ = (ba)^* \\
L_{2,3,3} &= b \cup b(\varepsilon \cup ab)^*(\varepsilon \cup ab) = b \cup b(ab)^* = b(ba)^* \\
L_{3,3,1} &= \emptyset \cup (\varepsilon \cup ab)(\varepsilon \cup ab)^*\emptyset = \emptyset \\
L_{3,3,2} &= a \cup (\varepsilon \cup ab)(\varepsilon \cup ab)^* a = a \cup (ab)^* a = (ab)^* a \\
L_{3,3,3} &= \varepsilon \cup ab \cup (\varepsilon \cup ab)(\varepsilon \cup ab)^*(\varepsilon \cup ab) = (ab)^*
\end{aligned}
$$

Die vom Automaten akzeptierte Sprache ist also
$L = \bigcup_{f \in F} L_f = L_{q_1,3,q_2} = a \cup (c \cup ab)(ab)^* a = (c \cup \varepsilon)a(ba)^*$

Aufgabe 7

Wir betrachten den endlichen Automaten über dem Alphabet $\Sigma = \{0, 1\}$ mit den Zuständen $\{A, B, C, D\}$. Startzustand ist A, einziger Endzustand D. Übergangstabelle ist

δ	A	B	C	D
0	A,B	D	A,D	B
1	C	A	A	C

Aufgabe 7

Wir betrachten den endlichen Automaten über dem Alphabet $\Sigma = \{0, 1\}$ mit den Zuständen $\{A, B, C, D\}$. Startzustand ist A, einziger Endzustand D. Übergangstabelle ist

δ	A	B	C	D
0	A,B	D	A,D	B
1	C	A	A	C

a) Zeichnen Sie den Automaten. Zeigen Sie, dass er nichtdeterministisch ist.

Aufgabe 7

Wir betrachten den endlichen Automaten über dem Alphabet $\Sigma = \{0, 1\}$ mit den Zuständen $\{A, B, C, D\}$. Startzustand ist A, einziger Endzustand D. Übergangstabelle ist

δ	A	B	C	D
0	A,B	D	A,D	B
1	C	A	A	C

a) Zeichnen Sie den Automaten. Zeigen Sie, dass er nichtdeterministisch ist.

b) Konstruieren Sie den zugehörigen deterministischen Automaten (Potenzmengenkonstruktion).

Aufgabe 7

Wir betrachten den endlichen Automaten über dem Alphabet $\Sigma = \{0, 1\}$ mit den Zuständen $\{A, B, C, D\}$. Startzustand ist A, einziger Endzustand D. Übergangstabelle ist

δ	A	B	C	D
0	A,B	D	A,D	B
1	C	A	A	C

a) Zeichnen Sie den Automaten. Zeigen Sie, dass er nichtdeterministisch ist.
b) Konstruieren Sie den zugehörigen deterministischen Automaten (Potenzmengenkonstruktion).
c) Entfernen Sie daraus alle eventuell vorkommenden nicht erreichbaren Zustände.

Aufgabe 7

Wir betrachten den endlichen Automaten über dem Alphabet $\Sigma = \{0, 1\}$ mit den Zuständen $\{A, B, C, D\}$. Startzustand ist A, einziger Endzustand D. Übergangstabelle ist

δ	A	B	C	D
0	A,B	D	A,D	B
1	C	A	A	C

a) Zeichnen Sie den Automaten. Zeigen Sie, dass er nichtdeterministisch ist.
b) Konstruieren Sie den zugehörigen deterministischen Automaten (Potenzmengenkonstruktion).
c) Entfernen Sie daraus alle eventuell vorkommenden nicht erreichbaren Zustände.
d) Konstruieren Sie daraus den zugehörigen Minimalautomaten.

Aufgabe 8a

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache Σ^* ist.

Aufgabe 8a

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache Σ^* ist.

Lösung:
Führe Tiefensuche beginnend beim Startzustand durch.

Aufgabe 8a

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache Σ^* ist.

Lösung:
Führe Tiefensuche beginnend beim Startzustand durch.
Kein nichtterminaler Zustand erreichbar, und DEA vollstaendig,

Aufgabe 8a

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache Σ^* ist.

Lösung:
Führe Tiefensuche beginnend beim Startzustand durch.
Kein nichtterminaler Zustand erreichbar, und DEA vollstaendig, dann Ausgabe: *Die dem Automaten zugehörige Sprache ist* Σ^*.

Aufgabe 8b

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache $\emptyset$ ist.

Aufgabe 8b

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache $\emptyset$ ist.

Lösung:
Führe Tiefensuche beginnend beim Startzustand durch.

Aufgabe 8b

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache $\emptyset$ ist.

Lösung:
Führe Tiefensuche beginnend beim Startzustand durch.
Kein terminaler Zustand erreichbar,

Aufgabe 8b

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache $\emptyset$ ist.

Lösung:
Führe Tiefensuche beginnend beim Startzustand durch.
Kein terminaler Zustand erreichbar,
dann Ausgabe: *Die dem Automaten zugehörige Sprache ist* $\emptyset$.

Aufgabe 8c

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache L_3 genau alle Wörter akzeptiert, die mit einer gegebenem Anfangszeichenfolge $w_1 w_2 \dots w_n$ beginnen.

Aufgabe 8c

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache L_3 genau alle Wörter akzeptiert, die mit einer gegebenem Anfangszeichenfolge $w_1 w_2 \ldots w_n$ beginnen.

Lösung:

Führe Breitensuche bis zur Tiefe n durch.

Aufgabe 8c

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache L_3 genau alle Wörter akzeptiert, die mit einer gegebenem Anfangszeichenfolge $w_1 w_2 \dots w_n$ beginnen.

Lösung:

Führe Breitensuche bis zur Tiefe n durch.

Die zugehörige Sprache ist höchstens L_3, wenn nur durch die Abarbeitung von $w_1 w_2 \dots w_n$ ein terminierender Zustand erreichbar ist.

Aufgabe 8c

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache L_3 genau alle Wörter akzeptiert, die mit einer gegebenem Anfangszeichenfolge $w_1 w_2 \ldots w_n$ beginnen.

Lösung:

Führe Breitensuche bis zur Tiefe n durch.

Die zugehörige Sprache ist höchstens L_3, wenn nur durch die Abarbeitung von $w_1 w_2 \ldots w_n$ ein terminierender Zustand erreichbar ist.

Bestimme Zustand p, der nach Abarbeitung von $w_1 w_2 \ldots w_n$ erreicht wird.

Aufgabe 8c

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache L_3 genau alle Wörter akzeptiert, die mit einer gegebenem Anfangszeichenfolge $w_1 w_2 \ldots w_n$ beginnen.

Lösung:

Führe Breitensuche bis zur Tiefe n durch.

Die zugehörige Sprache ist höchstens L_3, wenn nur durch die Abarbeitung von $w_1 w_2 \ldots w_n$ ein terminierender Zustand erreichbar ist.

Bestimme Zustand p, der nach Abarbeitung von $w_1 w_2 \ldots w_n$ erreicht wird.

Wende Algorithmus aus 8a) an.

Wenn Algorithmus aus 8a) erfolgreich terminiert.

Aufgabe 8c

Gegeben ist ein DEA (als Graph mit beschrifteten Knoten und Kanten). Schreiben Sie einen Algorithmus, der entscheidet, ob die zugehörige Sprache L_3 genau alle Wörter akzeptiert, die mit einer gegebenem Anfangszeichenfolge $w_1 w_2 \ldots w_n$ beginnen.

Lösung:

Führe Breitensuche bis zur Tiefe n durch.

Die zugehörige Sprache ist höchstens L_3, wenn nur durch die Abarbeitung von $w_1 w_2 \ldots w_n$ ein terminierender Zustand erreichbar ist.

Bestimme Zustand p, der nach Abarbeitung von $w_1 w_2 \ldots w_n$ erreicht wird.

Wende Algorithmus aus 8a) an.

Wenn Algorithmus aus 8a) erfolgreich terminiert.

dann Ausgabe: *Die dem Automaten zugehörige Sprache ist L_3.*

Definition: Nerode-Relation

Für eine Sprache $L \subseteq \Sigma^*$ ist die Nerode-Relation R_L definiert durch:

für $x, y \in \Sigma^*$ ist xR_Ly genau dann, wenn $(xz \in L \Leftrightarrow yz \in L)$ für alle $z \in \Sigma^*$ gilt.

Satz von Nerode:

Die folgenden Aussagen sind äquivalent:

Satz von Nerode:

Die folgenden Aussagen sind äquivalent:

$$L \subseteq \Sigma^* \text{ wird von einem DEA akzeptiert.}$$

Satz von Nerode:

Die folgenden Aussagen sind äquivalent:

$L \subseteq \Sigma^*$ wird von einem DEA akzeptiert.

$\Leftrightarrow$ L ist die Vereinigung von (einigen) Äquivalenzklassen einer rechtsinvarianten Äquivalenzrelation mit endlichem Index.

Satz von Nerode:

Die folgenden Aussagen sind äquivalent:

$L \subseteq \Sigma^*$ wird von einem DEA akzeptiert.

$\Leftrightarrow$ L ist die Vereinigung von (einigen) Äquivalenzklassen einer rechtsinvarianten Äquivalenzrelation mit endlichem Index.

$\Leftrightarrow$ Die Nerode-Relation hat endlichen Index.

Satz von Nerode:

Die folgenden Aussagen sind äquivalent:

$L \subseteq \Sigma^*$ wird von einem DEA akzeptiert.

$\Leftrightarrow$ L ist die Vereinigung von (einigen) Äquivalenzklassen einer rechtsinvarianten Äquivalenzrelation mit endlichem Index.

$\Leftrightarrow$ Die Nerode-Relation hat endlichen Index.

In andern Worten: Eine Sprache L ist genau dann regulär, wenn der Index von R_L endlich ist.

Satz:

Der Automat der Nerode-Relation ist minimal.

Aufgabe 1

Gib für die folgenden Sprachen jeweils die Äquivalenzklassen bezüglich der Nerode-Relation R_L an. Welche der Sprachen sind regulär? Konstruiere für diese Sprachen jeweils den Minimalautomaten.

Aufgabe 1

Gib für die folgenden Sprachen jeweils die Äquivalenzklassen bezüglich der Nerode-Relation R_L an. Welche der Sprachen sind regulär? Konstruiere für diese Sprachen jeweils den Minimalautomaten.

a) $L_1 = \{(ab)^i | i \in \mathbb{N}\}$

Aufgabe 1

Gib für die folgenden Sprachen jeweils die Äquivalenzklassen bezüglich der Nerode-Relation R_L an. Welche der Sprachen sind regulär? Konstruiere für diese Sprachen jeweils den Minimalautomaten.

a) $L_1 = \{(ab)^i | i \in \mathbb{N}\}$

b) $L_2 = \{a^i b^j | i, j \in \mathbb{N}\}$

Aufgabe 1

Gib für die folgenden Sprachen jeweils die Äquivalenzklassen bezüglich der Nerode-Relation R_L an. Welche der Sprachen sind regulär? Konstruiere für diese Sprachen jeweils den Minimalautomaten.

a) $L_1 = \{(ab)^i | i \in \mathbb{N}\}$

b) $L_2 = \{a^i b^j | i, j \in \mathbb{N}\}$

c) $L_3 = \{a^n b^n | n \in \mathbb{N}\}$

Aufgabe 2

Wieviele Zustände benötigt ein DEA, der alle durch 3 teilbaren Zahlen in Dezimaldarstellung akzeptiert, mindestens? Begründe deine Antwort. Bestimme die Äquivalenzklassen bezüglich der Nerode-Relation und konstruiere den Minimalautomaten.

Was haben wir bisher alles kennengelernt und können nun damit umgehen?

- Deterministische endliche Automaten

Was haben wir bisher alles kennengelernt und können nun damit umgehen?

- Deterministische endliche Automaten
- reguläre Ausdrücke

Was haben wir bisher alles kennengelernt und können nun damit umgehen?

- Deterministische endliche Automaten
- reguläre Ausdrücke
- Nichtdeterministische endliche Automaten

Was haben wir bisher alles kennengelernt und können nun damit umgehen?

- Deterministische endliche Automaten
- reguläre Ausdrücke
- Nichtdeterministische endliche Automaten
- NEA $\rightarrow$ DEA (Potenzmengenkonstruktion)

Was haben wir bisher alles kennengelernt und können nun damit umgehen?

- Deterministische endliche Automaten
- reguläre Ausdrücke
- Nichtdeterministische endliche Automaten
- NEA $\rightarrow$ DEA (Potenzmengenkonstruktion)
- ε-NEAs

Was haben wir bisher alles kennengelernt und können nun damit umgehen?

- Deterministische endliche Automaten
- reguläre Ausdrücke
- Nichtdeterministische endliche Automaten
- NEA $\rightarrow$ DEA (Potenzmengenkonstruktion)
- ε-NEAs
- Pumping-Lemma

Was haben wir bisher alles kennengelernt und können nun damit umgehen?

- Deterministische endliche Automaten
- reguläre Ausdrücke
- Nichtdeterministische endliche Automaten
- NEA $\rightarrow$ DEA (Potenzmengenkonstruktion)
- ε-NEAs
- Pumping-Lemma
- Minimalautomat / Äquivalenzklassenautomat

Was haben wir bisher alles kennengelernt und können nun damit umgehen?

- Deterministische endliche Automaten
- reguläre Ausdrücke
- Nichtdeterministische endliche Automaten
- NEA $\rightarrow$ DEA (Potenzmengenkonstruktion)
- ε-NEAs
- Pumping-Lemma
- Minimalautomat / Äquivalenzklassenautomat
- Nerode-Relation

Wahr oder falsch?

- Jede endliche Sprache ist regulär.

Wahr oder falsch?

- Jede endliche Sprache ist regulär.
- Die regulären Ausdrücke $b^*(ab^*)^*$ und $(a^* \cup b)^*$ sind äquivalent.

Wahr oder falsch?

- Jede endliche Sprache ist regulär.
- Die regulären Ausdrücke $b^*(ab^*)^*$ und $(a^* \cup b)^*$ sind äquivalent.
- Sei L eine reguläre Sprache. Dann existiert ein $n \in \mathbb{N}$ so, dass falls es ein Wort $w \in L$ mit $|w| \geq n$ gibt, w so in uvx mit $|uv| \leq n$ und $v \neq \varepsilon$ zerlegt werden kann, dass $uv^ix \in L$ ist für mindestens ein $i \in \mathbb{N}_0$.

Wahr oder falsch?

- Jede endliche Sprache ist regulär.
- Die regulären Ausdrücke $b^*(ab^*)^*$ und $(a^* \cup b)^*$ sind äquivalent.
- Sei L eine reguläre Sprache. Dann existiert ein $n \in \mathbb{N}$ so, dass falls es ein Wort $w \in L$ mit $|w| \geq n$ gibt, w so in uvx mit $|uv| \leq n$ und $v \neq \varepsilon$ zerlegt werden kann, dass $uv^ix \in L$ ist für mindestens ein $i \in \mathbb{N}_0$.
- Seien L_1, L_2 reguläre Sprachen, dann ist auch $L_1 \cap L_2$ regulär.

Bis zum nächsten Mal

Definition: deterministische Turingmaschine (DTM)

Eine deterministische Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, s, F)$ besteht aus:

Definition: deterministische Turingmaschine (DTM)

Eine deterministische Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, s, F)$ besteht aus:

Q, endliche Zustandsmenge

Definition: deterministische Turingmaschine (DTM)

Eine deterministische Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, s, F)$ besteht aus:

Q, endliche Zustandsmenge

Σ, endliches Bandalphabet

Definition: deterministische Turingmaschine (DTM)

Eine deterministische Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, s, F)$ besteht aus:

Q, endliche Zustandsmenge

Σ, endliches Bandalphabet

$\sqcup$, Blanksymbol, beachte $\sqcup \notin \Sigma$

Definition: deterministische Turingmaschine (DTM)

Eine deterministische Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, s, F)$ besteht aus:

Q, endliche Zustandsmenge

Σ, endliches Bandalphabet

$\sqcup$, Blanksymbol, beachte $\sqcup \notin \Sigma$

Γ, endliches Bandalphabet mit $\Sigma \cup \{\sqcup\} \subseteq \Gamma$

Definition: deterministische Turingmaschine (DTM)

Eine deterministische Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, s, F)$ besteht aus:

Q, endliche Zustandsmenge

Σ, endliches Bandalphabet

$\sqcup$, Blanksymbol, beachte $\sqcup \notin \Sigma$

Γ, endliches Bandalphabet mit $\Sigma \cup \{\sqcup\} \subseteq \Gamma$

$s \in Q$, einem Startzustand

Definition: deterministische Turingmaschine (DTM)

Eine deterministische Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, s, F)$ besteht aus:

- Q, endliche Zustandsmenge
- Σ, endliches Bandalphabet
- $\sqcup$, Blanksymbol, beachte $\sqcup \notin \Sigma$
- Γ, endliches Bandalphabet mit $\Sigma \cup \{\sqcup\} \subseteq \Gamma$
- $s \in Q$, einem Startzustand
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$, einer Überführungsfunktion.

Definition: deterministische Turingmaschine (DTM)

Eine deterministische Turingmaschine $M = (Q, \Sigma, \Gamma, \delta, s, F)$ besteht aus:

Q, endliche Zustandsmenge

Σ, endliches Bandalphabet

$\sqcup$, Blanksymbol, beachte $\sqcup \notin \Sigma$

Γ, endliches Bandalphabet mit $\Sigma \cup \{\sqcup\} \subseteq \Gamma$

$s \in Q$, einem Startzustand

$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$, einer Überführungsfunktion.

$F \subseteq Q$, Menge von Endzuständen. (Muss nicht sein.)

Definition:

- Eine Sprache $L \subseteq \Sigma^*$ heisst **rekursiv** oder **entscheidbar**, wenn es eine TM gibt, die auf allen Eingaben stoppt und eine Eingabe w genau dann akzeptiert, wenn $w \in L$ gilt.

Definition:

- Eine Sprache $L \subseteq \Sigma^*$ heisst **rekursiv** oder **entscheidbar**, wenn es eine TM gibt, die auf allen Eingaben stoppt und eine Eingabe w genau dann akzeptiert, wenn $w \in L$ gilt.
- Eine Sprache $L \subseteq \Sigma^*$ heisst **rekursiv-aufzählbar** oder **semi-entscheidbar**, wenn es eine TM gibt, die genau die Eingaben w akzeptiert für die $w \in L$. (Das Verhalten der TM für $w \notin L$ ist nicht defniniert.)

Definition:

- Eine Funktion $f : \Sigma^* \to \Gamma^*$ heisst **berechenbar** oder **totalrekursiv**, wenn es eine TM gibt, die bei Eingabe von $w \in \Sigma^*$ den Funktionswert $f(w) \in \Gamma^*$ ausgibt.

Definition:

- Eine Funktion $f : \Sigma^* \to \Gamma^*$ heisst **berechenbar** oder **totalrekursiv**, wenn es eine TM gibt, die bei Eingabe von $w \in \Sigma^*$ den Funktionswert $f(w) \in \Gamma^*$ ausgibt.
- Eine TM realisiert eine Funktion $f : \Sigma^* \to \Gamma^*$, falls gilt:
$$f(w) = \begin{cases} \text{Ausgabe der TM,} & \text{wenn sie bei Eingabe } w \text{ stoppt.} \\ \text{undefiniert,} & \text{sonst} \end{cases}$$

Definition:

- Eine Sprache $L \subseteq \Sigma^*$ ist **entscheidbar** $\Leftrightarrow$ ihre charakterische Funktion χ_L berechenbar ist, wobei gilt:

$$\chi_L : \Sigma^* \to \{0,1\} \text{ mit } \chi_L(w) = \begin{cases} 1, & \text{falls } w \in L \\ 0, & \text{sonst.} \end{cases}$$

Definition:

- Eine Sprache $L \subseteq \Sigma^*$ ist **entscheidbar** $\Leftrightarrow$ ihre charakterische Funktion χ_L berechenbar ist, wobei gilt:

$$\chi_L : \Sigma^* \to \{0,1\} \text{ mit } \chi_L(w) = \begin{cases} 1, & \text{falls } w \in L \\ 0, & \text{sonst.} \end{cases}$$

- Eine Sprache $L \subseteq \Sigma^*$ ist **semi-entscheidbar** $\Leftrightarrow$ Funktion χ_L^* berechenbar ist, wobei gilt:

$$\chi_L^*(w) = \begin{cases} 1, & \text{falls } w \in L \\ \text{undefiniert}, & \text{sonst.} \end{cases}$$

Churche These:

Die durch die formale Definition der Turing-Berechenbarkeit erfasste Klasse von Funktionen stimmt genau mit der Klasse der im intuitiven Sinne berechenbaren Funktionen überein.

Aufgabe 3

Entwerfe eine DTM, die als Eingabe eine (zusammenhängende) Folge von Nullen erhält und als Ausgabe eine Folge von Einsen der doppelten Länge auf das Band schreibt. Beschreibe zunächst kurz in Worten, wie die Turingmaschine funktioniert.

Aufgabe 4

Entwerfe eine DTM, die für einen Binärstring mit mindestens 5 Zeichen, das vierte Zeichen von rechts ausgibt. Beschreibe zunächst kurz in Worten, wie die Turingmaschine funktioniert.

Reflexion

Was haben wir heute gelernt?

Reflexion

Was haben wir heute gelernt?

- Nerode Relation

Reflexion

Was haben wir heute gelernt?

- Nerode Relation
- Pumping-Lemma wiederholt

Reflexion

Was haben wir heute gelernt?

- Nerode Relation
- Pumping-Lemma wiederholt
- Kapitel reguläre Sprachen abgeschlossen

Reflexion

Was haben wir heute gelernt?

- Nerode Relation
- Pumping-Lemma wiederholt
- Kapitel reguläre Sprachen abgeschlossen
- deterministische Turingmaschine

Noch Fragen?

Vorschau

Vorschau

- Das Postsche Korrespondenzproblem

Vorschau

- Das Postsche Korrespondenzproblem
- mehr zu Turingmaschinen

Vorschau

- Das Postsche Korrespondenzproblem
- mehr zu Turingmaschinen
- Entscheidbarkeit, Semientscheidbarkeit

Vorschau

- Das Postsche Korrespondenzproblem
- mehr zu Turingmaschinen
- Entscheidbarkeit, Semientscheidbarkeit
- Abschlusseigenschaften

Bis zum nächsten Mal