Grundbegriffe der Informatik
WS 2011/12
Tutorium in der Woche 13

Gehalten in den Tutorien Nr. 10, Nr. 14

Philipp Basler (philippbasler@googlemail.com)
Nils Braun (area51.nils@googlemail.com)

KIT - Karlsruher Institut für Technologie

30.01.2012 & 31.01.2012

Inhaltsverzeichnis

Informationen zum nächsten Blatt

Blatt Nr. 13

Abgabetermin	3.2.2012 um 12:30
Abgabeort	Briefkasten
Themen	Äquivalenzrelationen, Nerode-Ä und Funktionen
Maximale Punkte	18

Häufige Fehler auf dem letzten Übungsblatt

Blatt Nr. 12

- Aufgabe 3: Denkt an die Grenzfälle!

Was bleibt?

Was kann eine Turingmaschine?

Was bleibt?

Was kann eine Turingmaschine?

Band, Schreibkopf, Zustände, Übergangs usw.

Was bleibt?

Was ist eine Äquivalenzrelation?

Was bleibt?

Was ist eine Äquivalenzrelation?

symmetrisch, reflexiv, transitiv

Was bleibt?

Wie zeige ich Mengengleichheit?

Was bleibt?

Wie zeige ich Mengengleichheit?

Element aus der einen Menge liegt in der anderen.

Was bleibt?

Hat jeder Automat einen Startzustand?

Was bleibt?

Hat jeder Automat einen Startzustand?

Ja (bis auf den leeren Automat)

Wir betrachten zuerst nur Turingmaschinen, die bei jedem Eingabewort halten!

Zeitkomplexität

Definition

Die Zeitkomplexität Time(n) einer Turingmaschine ist die maximale Anzahl an Schritten, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case).

Zeitkomplexität

Definition

Die Zeitkomplexität Time(n) einer Turingmaschine ist die maximale Anzahl an Schritten, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case).

Beispiel: Überprüfung auf Palindrom:

- Erstes Symbol -> letztes Symbol (n)

Zeitkomplexität

Definition

Die Zeitkomplexität Time(n) einer Turingmaschine ist die maximale Anzahl an Schritten, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case).

Beispiel: Überprüfung auf Palindrom:

- Erstes Symbol -> letztes Symbol (n)
- Zurück zum ersten (n)

Zeitkomplexität

Definition

Die Zeitkomplexität Time(n) einer Turingmaschine ist die maximale Anzahl an Schritten, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case).

Beispiel: Überprüfung auf Palindrom:

- Erstes Symbol -> letztes Symbol (n)
- Zurück zum ersten (n)
- Mit kürzerem Wort wiederholen ($1 + T(n-2)$)

Zeitkomplexität

Definition

Die Zeitkomplexität Time(n) einer Turingmaschine ist die maximale Anzahl an Schritten, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case).

Beispiel: Überprüfung auf Palindrom:

- Erstes Symbol -> letztes Symbol (n)
- Zurück zum ersten (n)
- Mit kürzerem Wort wiederholen ($1 + T(n-2)$)

Zeitkomplexität

Definition

Die Zeitkomplexität Time(n) einer Turingmaschine ist die maximale Anzahl an Schritten, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case).

Beispiel: Überprüfung auf Palindrom:

- Erstes Symbol -> letztes Symbol (n)
- Zurück zum ersten (n)
- Mit kürzerem Wort wiederholen ($1 + T(n-2)$)

Also insgesamt $T(n) \leq n + 1 + n + T(n-2)$ bzw.

$$T(n) - T(n-2) \leq 2n + 1 \in O(n)$$

Daraus folgern wir

$$T(n) \in O(n^2)$$

Platzkomplexität

Definition

Die Platzkomplexität Space(n) einer Turingmaschine ist die maximale Anzahl an Feldern, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case). Benötigt wird ein Feld, wenn es vom Schreibkopf besucht wird.

Platzkomplexität

Definition

Die Platzkomplexität Space(n) einer Turingmaschine ist die maximale Anzahl an Feldern, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case). Benötigt wird ein Feld, wenn es vom Schreibkopf besucht wird.

Beispiel: Überprüfung auf Palindrom:

- Erstes Symbol -> letztes Symbol ($n + 1$)

Platzkomplexität

Definition

Die Platzkomplexität Space(n) einer Turingmaschine ist die maximale Anzahl an Feldern, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case). Benötigt wird ein Feld, wenn es vom Schreibkopf besucht wird.

Beispiel: Überprüfung auf Palindrom:

- Erstes Symbol -> letztes Symbol ($n + 1$)
- Zurück zum ersten (0)

Platzkomplexität

Definition

Die Platzkomplexität Space(n) einer Turingmaschine ist die maximale Anzahl an Feldern, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case). Benötigt wird ein Feld, wenn es vom Schreibkopf besucht wird.

Beispiel: Überprüfung auf Palindrom:

- Erstes Symbol -> letztes Symbol ($n + 1$)
- Zurück zum ersten (0)
- Mit kürzerem Wort wiederholen (0)

Platzkomplexität

Definition

Die Platzkomplexität Space(n) einer Turingmaschine ist die maximale Anzahl an Feldern, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case). Benötigt wird ein Feld, wenn es vom Schreibkopf besucht wird.

Beispiel: Überprüfung auf Palindrom:

- Erstes Symbol -> letztes Symbol ($n + 1$)
- Zurück zum ersten (0)
- Mit kürzerem Wort wiederholen (0)

Platzkomplexität

Definition

Die Platzkomplexität Space(n) einer Turingmaschine ist die maximale Anzahl an Feldern, die eine Turingmaschine bei Eingabe eines Worts der Länge n benötigen kann (worst-case). Benötigt wird ein Feld, wenn es vom Schreibkopf besucht wird.

Beispiel: Überprüfung auf Palindrom:

- Erstes Symbol -> letztes Symbol ($n + 1$)
- Zurück zum ersten (0)
- Mit kürzerem Wort wiederholen (0)

Also insgesamt

$$n + 1$$

Definition

- **P** ist die Menge aller Entscheidungsprobleme, die von Turingmaschinen entschieden werden können, deren Zeitkomplexität polynomiell ist.
- **PSPACE** ist die Menge aller Entscheidungsprobleme, die von Turingmaschinen entschieden werden können, deren Raumkomplexität polynomiell ist.

Definition

- **P** ist die Menge aller Entscheidungsprobleme, die von Turingmaschinen entschieden werden können, deren Zeitkomplexität polynomiell ist.
- **PSPACE** ist die Menge aller Entscheidungsprobleme, die von Turingmaschinen entschieden werden können, deren Raumkomplexität polynomiell ist.

Es ist

$$\mathbf{P} \subset \mathbf{PSPACE}$$

Definition

- **P** ist die Menge aller Entscheidungsprobleme, die von Turingmaschinen entschieden werden können, deren Zeitkomplexität polynomiell ist.
- **PSPACE** ist die Menge aller Entscheidungsprobleme, die von Turingmaschinen entschieden werden können, deren Raumkomplexität polynomiell ist.

Es ist

$$\mathbf{P} \subset \mathbf{PSPACE}$$

Mit t Schritten kann ich maximal $t+1$ Felder erreichen.

Aufgabe (WS 2011)

Die Turingmaschine T mit Anfangszustand S sei durch folgende Überführungsfunktion gegeben

	S	S_a	S_b	R
a	$(X, S_a, -1)$	$(a, S_a, -1)$	$(a, S_b, -1)$	$(a, R, 1)$
b	$(X, S_b, -1)$	$(b, S_a, -1)$	$(b, S_b, -1)$	$(b, R, 1)$
X	$(X, S, 1)$	$(X, S_a, -1)$	$(X, S_b, -1)$	$(X, S, 1)$
$\square$	-	$(a, R, 1)$	$(b, R, 1)$	-

Was steht bei der Eingabe des Wortes $w \in \{a, b\}^*$ am Ende der Berechnung auf dem Band?
Welche Platzkomplexität hat T? (Exakte Angabe in Abhängigkeit von der Länge der Eingabe!)
Geben Sie eine einfache Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ an, so dass die Zeitkomplexität von T in $\Theta(f(n))$ liegt.

Aufgabe (WS 2011)

Die Turingmaschine T mit Anfangszustand S sei durch folgende Überführungsfunktion gegeben

	S	S_a	S_b	R
a	$(X, S_a, -1)$	$(a, S_a, -1)$	$(a, S_b, -1)$	$(a, R, 1)$
b	$(X, S_b, -1)$	$(b, S_a, -1)$	$(b, S_b, -1)$	$(b, R, 1)$
X	$(X, S, 1)$	$(X, S_a, -1)$	$(X, S_b, -1)$	$(X, S, 1)$
$\square$	-	$(a, R, 1)$	$(b, R, 1)$	-

Was steht bei der Eingabe des Wortes $w \in \{a, b\}^*$ am Ende der Berechnung auf dem Band?

Lösung: Am Ende steht das Wort $R(w)\mathtt{X}^{|w|}$ auf dem Band, wobei $R(w)$ das Spiegelbild von w ist.

Welche Platzkomplexität hat T? (Exakte Angabe in Abhängigkeit von der Länge der Eingabe!)

Geben Sie eine einfache Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ an, so dass die Zeitkomplexität von T in $\Theta(f(n))$ liegt.

Aufgabe (WS 2011)

Die Turingmaschine T mit Anfangszustand S sei durch folgende Überführungsfunktion gegeben

	S	S_a	S_b	R
a	$(X, S_a, -1)$	$(a, S_a, -1)$	$(a, S_b, -1)$	$(a, R, 1)$
b	$(X, S_b, -1)$	$(b, S_a, -1)$	$(b, S_b, -1)$	$(b, R, 1)$
X	$(X, S, 1)$	$(X, S_a, -1)$	$(X, S_b, -1)$	$(X, S, 1)$
$\square$	-	$(a, R, 1)$	$(b, R, 1)$	-

Was steht bei der Eingabe des Wortes $w \in \{a, b\}^*$ am Ende der Berechnung auf dem Band?

Welche Platzkomplexität hat T? (Exakte Angabe in Abhängigkeit von der Länge der Eingabe!)

Lösung: Eingabe der Länge n: Platzbedarf ist $2n + 1$

Geben Sie eine einfache Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ an, so dass die Zeitkomplexität von T in $\Theta(f(n))$ liegt.

Aufgabe (WS 2011)

Die Turingmaschine T mit Anfangszustand S sei durch folgende Überführungsfunktion gegeben

	S	S_a	S_b	R
a	$(X, S_a, -1)$	$(a, S_a, -1)$	$(a, S_b, -1)$	$(a, R, 1)$
b	$(X, S_b, -1)$	$(b, S_a, -1)$	$(b, S_b, -1)$	$(b, R, 1)$
X	$(X, S, 1)$	$(X, S_a, -1)$	$(X, S_b, -1)$	$(X, S, 1)$
$\square$	-	$(a, R, 1)$	$(b, R, 1)$	-

Was steht bei der Eingabe des Wortes $w \in \{a, b\}^*$ am Ende der Berechnung auf dem Band?
Welche Platzkomplexität hat T? (Exakte Angabe in Abhängigkeit von der Länge der Eingabe!)
Geben Sie eine einfache Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ an, so dass die Zeitkomplexität von T in $\Theta(f(n))$ liegt. *Lösung*: Eingabe der Länge n: Zeitbedarf in $\Theta(n^2)$

Unentscheidbare Probleme

Es existieren Probleme, die von keiner Turingmaschine entschieden werden können.

Unentscheidbare Probleme

Es existieren Probleme, die von keiner Turingmaschine entschieden werden können. Also nicht aufgrund Platz- oder Zeitgründen sondern grundsätzlich nicht.

Unentscheidbare Probleme

Es existieren Probleme, die von keiner Turingmaschine entschieden werden können. Also nicht aufgrund Platz- oder Zeitgründen sondern grundsätzlich nicht. Entscheidbar heißt, dass es eine Turingmaschine gibt, die für jede Eingabe hält und entscheiden kann, ob das Wort in der Sprache liegt oder nicht. Statt Sprachen spricht man auch gerne von Problemen.

Codierung von Turingmaschinen

Satz

Es existiert eine universelle Turingmaschine U, die für zwei Eingaben $[\mathtt{w}_1][\mathtt{w}_2]$

- *überprüft ob $\mathtt{w}_1$ eine Turingmaschine T codiert*
- *falls ja, die Eingabe $\mathtt{w}_2$ auf dieser Turingmaschine simuliert*
- *Das Ergebnis davon präsentiert (falls T hält)*

Halteproblem

Satz

Es ist nicht möglich eine Turingmaschine H zu bauen, die für jede Turingmaschine T und jede Eingabe w entscheidet, ob T bei der Eingabe von w hält.

Beweis

Sei H solch eine Turingmaschine, die bei Eingabe einer Turingmaschine und eines Wortes entscheidet, ob die Turingmaschine hält (1) oder nicht (0). Dann existieren Turingmaschinen G und F mit:

- Wenn T nicht hält, dann hält H mit 0 und G mit 0.
- Wenn T hält, dann hält H mit 1 und G nicht.
- F ersetzt die Eingabe x mit xx und ruft G auf.

Widerspruch

Was passiert jetzt, wenn F mit seiner eigenen Codierung $c(F)$ aufgerufen wird?

Widerspruch

Was passiert jetzt, wenn F mit seiner eigenen Codierung $c(F)$ aufgerufen wird?

- Angenommen F hält mit der Eingabe $c(F)$. Dann sagt die Supermaschine H 1 und hält bei der Eingabe $c(F)c(F)$. Dann hält G nicht. Und damit F auch nicht.

Widerspruch

Was passiert jetzt, wenn F mit seiner eigenen Codierung $c(F)$ aufgerufen wird?

- Angenommen F hält mit der Eingabe $c(F)$. Dann sagt die Supermaschine H 1 und hält bei der Eingabe $c(F)c(F)$. Dann hält G nicht. Und damit F auch nicht.
- Angenommen F hält mit der EIngabe $c(F)$ nicht. Dann sagt H 0. Und damit hält G und somit auch F.

Widerspruch

Was passiert jetzt, wenn F mit seiner eigenen Codierung $c(F)$ aufgerufen wird?

- Angenommen F hält mit der Eingabe $c(F)$. Dann sagt die Supermaschine H 1 und hält bei der Eingabe $c(F)c(F)$. Dann hält G nicht. Und damit F auch nicht.
- Angenommen F hält mit der EIngabe $c(F)$ nicht. Dann sagt H 0. Und damit hält G und somit auch F.

Widerspruch

Was passiert jetzt, wenn F mit seiner eigenen Codierung $c(F)$ aufgerufen wird?

- Angenommen F hält mit der Eingabe $c(F)$. Dann sagt die Supermaschine H 1 und hält bei der Eingabe $c(F)c(F)$. Dann hält G nicht. Und damit F auch nicht.
- Angenommen F hält mit der EIngabe $c(F)$ nicht. Dann sagt H 0. Und damit hält G und somit auch F.

Also hält F genau dann, wenn F nicht hält.

Beweis Möglichkeit 2:

Sei eine Tabelle x_i, f_j gegeben, wobei die x_i alle Codierungen einer Turingmaschine sind und die f_j die berechneten Funktionen der Turingmaschine T_j sind. Sei jetzt H solch eine Supermaschine und G wieder die Maschine, die

- Wenn H mitteilt, dass $T_{x_i}(x_i)$ hält, dann geht G in eine Endlosschleife.
- Wenn H mitteilt, dass $T_{x_i}(x_i)$ nicht hält, dann hält G (und liefert irgendein Ergebnis, etwa 0).

Jede mögliche Turingmaschine T_{x_i} verhält sich also für eine bestimmte Eingabe x_i genau anders wie G. Also ist G eine Turingmaschine, die nicht in allen Turingmaschinen liegt.

Definition

Ein fleißiger Bieber ist eine Turingmaschine, die $n + 1$ Zustände hat, wobei ein Anfangszustand und ein Haltezustand darunter sind und die nur Einsen produzieren kann.

Definition

Ein fleißiger Bieber ist eine Turingmaschine, die $n+1$ Zustände hat, wobei ein Anfangszustand und ein Haltezustand darunter sind und die nur Einsen produzieren kann.

Definition

Als Busy-Beaver-Funktion $bb(n)$ wird die maximale Anzahl an Einsen bezeichnet, die ein fleißiger Bieber mit $n+1$ Zuständen auf dem Band hinterlassen kann.

Was ist $bb(4)$?

Was ist $bb(4)$? Wer kann mehr Einser zeichnen?

Was ist $bb(4)$? Wer kann mehr Einser zeichnen?

	A	B	C	D
□	1, R, B	1, L, A	1, R, H	1, R, D
1	1, L, B	□, L, C	1, L, D	□, R, A

Satz

Die Busy-Beaver-Funktion ist nicht berechenbar (es gibt keine Turingmaschine, die die Funktionswerte als Ausgabe liefert).

Äquivalenz. . .

Definition

Eine Relation R nennt man **Äquivalenzrelation** wenn sie folgende Eigenschaften erfüllt:

- symmetrisch
- reflexiv
- transitiv

Äquivalenz. . .

Definition

Eine Relation R nennt man **Äquivalenzrelation** wenn sie folgende Eigenschaften erfüllt:

- symmetrisch
- reflexiv
- transitiv

Definition

Sind zwei Elemente $(x, y) \in R$, so schreibt man auch xRy. Alle Elemente, die miteinander in Relation stehen, befinden sich in der selben **Äquivalenzklasse**:

$$[x]_R = \{y \mid yRx\}$$

Faktormenge

Definition

Die Menge aller Äquivalenzklassen einer Menge M zur Relation R bzeichnet man als **Faktormenge** und schreibt $M_{/R}$.

Faktormenge

Definition

Die Menge aller Äquivalenzklassen einer Menge M zur Relation R bzeichnet man als **Faktormenge** und schreibt $M_{/R}$.

Zeichnung an der Tafel!

Beweisen Sie...

- Aus xRy folgt $[x]_R = [y]_R$
- Existiert ein $z \in [x]_R$ und $z \in [y]_R$, so ist $[x]_R = [y]_R$
- Zu $R = \textbf{mod } 6$ gibt es 6 Äquivalenzklassen.

Nerode-Äquivalenzrelation

Definition

Sei $L \subseteq A^*$ eine formale Sprache. Definiere für zwei Wörter $w_1, w_2 \in A^*$:

$$w_1 \equiv_L w_2 \iff (\forall w \in A^* : w_1 w \in L \iff w_2 w \in L)$$

Die Relation $\equiv_L$ nennt man **Äquivalenzrelation von Nerode**

Nerode-Äquivalenzrelation

Definition

Sei $L \subseteq A^*$ eine formale Sprache. Definiere für zwei Wörter $w_1, w_2 \in A^*$:

$$w_1 \equiv_L w_2 \iff (\forall w \in A^* : w_1 w \in L \iff w_2 w \in L)$$

Die Relation $\equiv_L$ nennt man **Äquivalenzrelation von Nerode**

Äh, hä?

Ein Beispiel

Sei $L \subset A^*$ die Sprache der Wörter, die das Teilwort $\mathtt{ba}$ nicht enthalten.

Ein Beispiel

Sei $L \subset A^*$ die Sprache der Wörter, die das Teilwort $\mathtt{ba}$ nicht enthalten.

$$L = \langle \mathtt{a}^* \mathtt{b}^* \rangle$$

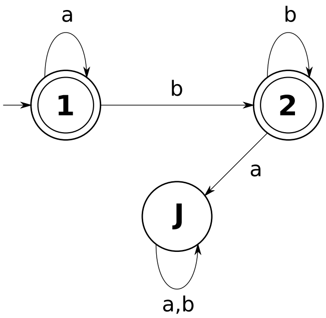
Wie sieht ein endlicher Automat dazu aus?

Ein Beispiel

Sei $L \subset A^*$ die Sprache der Wörter, die das Teilwort $\mathtt{ba}$ nicht enthalten.

$$L = \langle \mathtt{a}^* \mathtt{b}^* \rangle$$

Wie sieht ein endlicher Automat dazu aus?



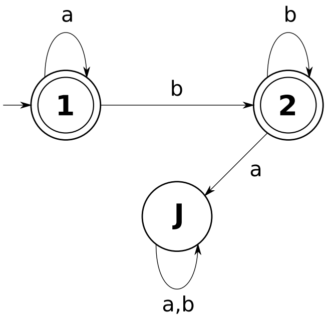
Was sind die Nerode Äquivalenzklassen?

Ein Beispiel

Sei $L \subset A^*$ die Sprache der Wörter, die das Teilwort ba nicht enthalten.

$$L = \langle \mathtt{a}^* \mathtt{b}^* \rangle$$

Wie sieht ein endlicher Automat dazu aus?



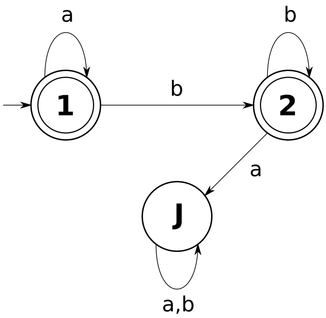
Was sind die Nerode Äquivalenzklassen? Wie komme ich in Zustand 1, 2, J?

Ein Beispiel

Sei $L \subset A^*$ die Sprache der Wörter, die das Teilwort $\mathtt{ba}$ nicht enthalten.

$$L = \langle \mathtt{a}^*\mathtt{b}^* \rangle$$

Wie sieht ein endlicher Automat dazu aus?



Was sind die Nerode Äquivalenzklassen? Wie komme ich in Zustand 1, 2, J?

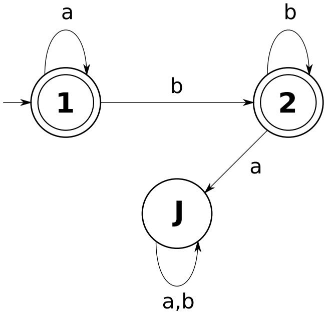
$$\mathtt{a}^*, \mathtt{a}^*\mathtt{bb}^*, \mathtt{a}^*\mathtt{bb}^*\mathtt{a}\{\mathtt{a}, \mathtt{b}\}^*$$

Ein Beispiel

Sei $L \subset A^*$ die Sprache der Wörter, die das Teilwort $\mathtt{ba}$ nicht enthalten.

$$L = \langle \mathtt{a}^*\mathtt{b}^* \rangle$$

Wie sieht ein endlicher Automat dazu aus?



Was sind die Nerode Äquivalenzklassen? Wie komme ich in Zustand 1, 2, J?

$$\mathtt{a}^*, \mathtt{a}^*\mathtt{b}\mathtt{b}^*, \mathtt{a}^*\mathtt{b}\mathtt{b}^*\mathtt{a}\{\mathtt{a}, \mathtt{b}\}^*$$

Wähle Vertreter!

$$[\varepsilon], [\mathtt{b}], [\mathtt{ba}]$$

Noch ein Beispiel

Sei $L \subset A^*$ die Sprache

$$L = \{\mathtt{a}^n\mathtt{b}\mathtt{a}^n \mid n \in \mathbb{N}\}$$

Wie sieht ein endlicher Automat dazu aus?

Noch ein Beispiel

Sei $L \subset A^*$ die Sprache

$$L = \{\mathtt{a}^n\mathtt{ba}^n \mid n \in \mathbb{N}\}$$

Wie sieht ein endlicher Automat dazu aus?

Argh! Es gibt keinen!

Noch ein Beispiel

Sei $L \subset A^*$ die Sprache

$$L = \{\mathtt{a}^n\mathtt{ba}^n \mid n \in \mathbb{N}\}$$

Wie sieht ein endlicher Automat dazu aus?

Was sind die Nerode Äquivalenzklassen?

Argh! Es gibt keinen!

Noch ein Beispiel

Sei $L \subset A^*$ die Sprache

$$L = \{\mathtt{a}^n\mathtt{b}\mathtt{a}^n \mid n \in \mathbb{N}\}$$

Wie sieht ein endlicher Automat dazu aus?

Was sind die Nerode Äquivalenzklassen? Was passiert bei

- $\mathtt{a}^i, i \in \mathbb{N}$

Argh! Es gibt keinen!

Noch ein Beispiel

Sei $L \subset A^*$ die Sprache

$$L = \{\mathtt{a}^n\mathtt{ba}^n \mid n \in \mathbb{N}\}$$

Wie sieht ein endlicher Automat dazu aus?

Was sind die Nerode Äquivalenzklassen? Was passiert bei

- $\mathtt{a}^i, i \in \mathbb{N}$
- $\mathtt{a}^i\mathtt{b}, i \in \mathbb{N}$

Argh! Es gibt keinen!

Noch ein Beispiel

Sei $L \subset A^*$ die Sprache

$$L = \{\mathtt{a}^n\mathtt{b}\mathtt{a}^n \mid n \in \mathbb{N}\}$$

Wie sieht ein endlicher Automat dazu aus?

Argh! Es gibt keinen!

Was sind die Nerode Äquivalenzklassen? Was passiert bei

- $\mathtt{a}^i, i \in \mathbb{N}$
- $\mathtt{a}^i\mathtt{b}, i \in \mathbb{N}$
- dem Rest

Noch ein Beispiel

Sei $L \subset A^*$ die Sprache

$$L = \{\mathtt{a}^n\mathtt{b}\mathtt{a}^n \mid n \in \mathbb{N}\}$$

Wie sieht ein endlicher Automat dazu aus?

Argh! Es gibt keinen!

Was sind die Nerode Äquivalenzklassen? Was passiert bei

- $\mathtt{a}^i, i \in \mathbb{N}$
- $\mathtt{a}^i\mathtt{b}, i \in \mathbb{N}$
- dem Rest

Noch ein Beispiel

Sei $L \subset A^*$ die Sprache

$$L = \{\mathrm{a}^n\mathrm{b}\mathrm{a}^n \mid n \in \mathbb{N}\}$$

Wie sieht ein endlicher Automat dazu aus?

Argh! Es gibt keinen!

Was sind die Nerode Äquivalenzklassen? Was passiert bei

- $\mathrm{a}^i, i \in \mathbb{N}$
- $\mathrm{a}^i\mathrm{b}, i \in \mathbb{N}$
- dem Rest

$$\{[\mathrm{a}^i] \mid i \in \mathbb{N}\} \cup \{[\mathrm{a}^i\mathrm{b}] \mid i \in \mathbb{N}\} \cup [\mathrm{ba}]$$

Was ihr nun wissen solltet

- Wie man Zeit- und Platzbedarf berechnet
- Wie man die Nerode-Äquivalenzklassen bestimmt.
- Wann die Übungsklausur ist.

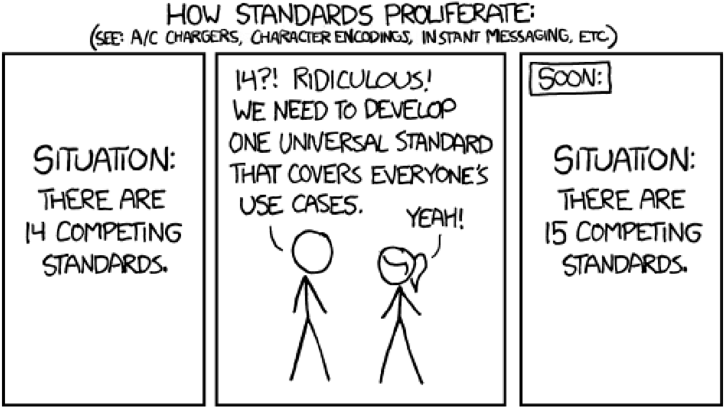


Abbildung: `http://www.xkcd.com`

Kontakt via E-Mail an Philipp Basler oder Nils Braun
gbi.ugroup.hostzi.com