Karlsruhe Institute of Technology

Grundbegriffe der Informatik - Tutorium 21

Christian Jülg
Wintersemester 2012/13
29. Januar 2013

http://gbi-tutor.blogspot.com

Übersicht

Übersicht

Guten Morgen...

Aufgabenblatt 12

Aufgabenblatt 13

Probeklausur

WDH: Turingmaschine

Unentscheidbare Probleme

Äquivalenzrelationen

Halbordnungen

Ordnungen

Abschluss

Zum Warmwerden...

Eine Turingmaschine...

1. ... kann eine Typ-1 Sprache realisieren.
2. ... ist weniger mächtig als ein endlicher Automat
3. ... befindet sich stets in einer Konfiguration.

Eine TM entscheidet eine Sprache L...

1. ... gdw sie L akzeptiert.
2. ... gdw sie auf alle Eingaben stoppt.
3. ... wenn sich für L ein Akzeptor angeben lässt.

Eine Funktion f ist berechenbar ...

1. ... wenn ein Algorithmus existiert, der f berechnet.
2. ... wenn eine TM diese Funktion realisiert.
3. ... die TM für alle Eingaben terminiert.

Zum Warmwerden...

Eine Turingmaschine...

1. ... kann eine Typ-1 Sprache realisieren.
2. ... ist weniger mächtig als ein endlicher Automat
3. ... befindet sich stets in einer Konfiguration.

Eine TM entscheidet eine Sprache L...

1. ... gdw sie L akzeptiert.
2. ... gdw sie auf alle Eingaben stoppt.
3. ... wenn sich für L ein Akzeptor angeben lässt.

Eine Funktion f ist berechenbar ...

1. ... wenn ein Algorithmus existiert, der f berechnet.
2. ... wenn eine TM diese Funktion realisiert.
3. ... die TM für alle Eingaben terminiert.

Zum Warmwerden...

Eine Turingmaschine...

1. ... kann eine Typ-1 Sprache realisieren.
2. ... ist weniger mächtig als ein endlicher Automat
3. ... befindet sich stets in einer Konfiguration.

Eine TM entscheidet eine Sprache L...

1. ... gdw sie L akzeptiert.
2. ... gdw sie auf alle Eingaben stoppt.
3. ... wenn sich für L ein Akzeptor angeben lässt.

Eine Funktion f ist berechenbar ...

1. ... wenn ein Algorithmus existiert, der f berechnet.
2. ... wenn eine TM diese Funktion realisiert.
3. ... die TM für alle Eingaben terminiert.

Zum Warmwerden...

Eine Turingmaschine...

1. ... kann eine Typ-1 Sprache realisieren.
2. ... ist weniger mächtig als ein endlicher Automat
3. ... befindet sich stets in einer Konfiguration.

Eine TM entscheidet eine Sprache L...

1. ... gdw sie L akzeptiert.
2. ... gdw sie auf alle Eingaben stoppt.
3. ... wenn sich für L ein Akzeptor angeben lässt.

Eine Funktion f ist berechenbar ...

1. ... wenn ein Algorithmus existiert, der f berechnet.
2. ... wenn eine TM diese Funktion realisiert.
3. ... die TM für alle Eingaben terminiert.

Übersicht

Aufgabenblatt 12

Blatt 12

- Abgaben: 10 / 19
- Punkte: Durchschnitt 8,25 von 19

häufige Fehler:

- 3) Vorgehensweise der TM in eigenen Worten heißt:
  grob die Positionswechsel und Schreibvorgänge beschreiben, aber nicht einzelne Kanten des TM-Graphen oder Einträge der TM-Tabelle beschreiben

Übersicht

Guten Morgen...

Aufgabenblatt 12

Aufgabenblatt 13

Probeklausur

WDH: Turingmaschine

Unentscheidbare Probleme

Äquivalenzrelationen

Halbordnungen

Ordnungen

Abschluss

Aufgabenblatt 13

Blatt 13

- Abgabe: 01.02.2013 um 12:30 Uhr im Untergeschoss des Infobaus
- Punkte: maximal 22

Themen

- Akzeptoren
- Äquivalenzrelationen
- Nerode Relationen

Übersicht

Guten Morgen...

Aufgabenblatt 12

Aufgabenblatt 13

Probeklausur

WDH: Turingmaschine

Unentscheidbare Probleme

Äquivalenzrelationen

Halbordnungen

Ordnungen

Abschluss

Probeklausur

Termin

- Freitag den 01.02.2013 anstelle der Gbi-Übung
- Ort: je nach Matrikelnummer im Audimax oder im HS -101 oder -102 im Geb. 50.34
- Teilnahme freiwillig
- Tutoren korrigieren die Abgaben, Rückgabe nächste Woche im Tutorium
- Ergebnis dient nur eurer Selbsteinschätzung
- Aufgaben werden von Tutoren erstellt, keine Garantie auf identische Aufgaben in der Klausur

Übersicht

Definition der TM

Ganz genau

Eine Turingmaschine $T = (Z, z_0, X, f, g, m)$ ist festgelegt durch

- eine endlichen **Zustandsmenge** Z
- einen **Anfangszustand** $z_0 \in Z$
- ein endliches **Bandalphabet** X

Definition der TM

Karlsruhe Institute of Technology

Ganz genau

Eine Turingmaschine $T = (Z, z_0, X, f, g, m)$ ist festgelegt durch

- eine endlichen **Zustandsmenge** Z
- einen **Anfangszustand** $z_0 \in Z$
- ein endliches **Bandalphabet** X
- eine partielle **Zustandsüberführung**sfunktion $f : Z \times X \dashrightarrow Z$
- eine partielle **Ausgabe**funktion $g : Z \times X \dashrightarrow X$ und
- eine partielle **Bewegung**sfunktion $m : Z \times X \dashrightarrow \{-1, 0, 1\}$

Definiton der TM

Anmerkungen

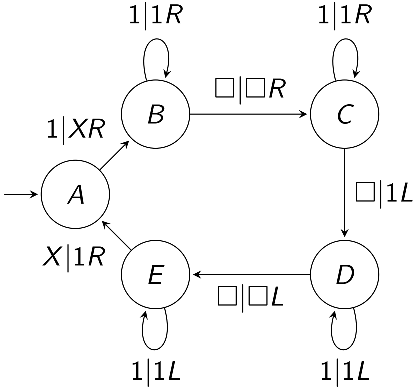
- Die Funktionen **f**, **g** und **m** beschreiben zusammen, wie das aktuell eingelesene Zeichen verarbeitet werden soll (haben gemeinsamen Definitionsbereich).

Definiton der TM

Anmerkungen

- Die Funktionen **f**, **g** und **m** beschreiben zusammen, wie das aktuell eingelesene Zeichen verarbeitet werden soll (haben gemeinsamen Definitionsbereich).
- Bei der Bewegungsfunktion bedeutet **-1 oder L** eine Bewegung des Lese-/Schreibkopfes nach links, **1 oder R** eine Bewegung nach rechts und **0 oder N** ein Stehenbleiben.

Bekanntes Beispiel



	A	B	C	D	E
□		□,R,C	1,L,D	□,L,E	
1	*X*, *R*, *B*	1,R,B	1,R,C	1,L,D	1,L,E
X					1,R,A

Konfiguration einer TM

Eine Turingmaschine befindet sich zu jedem Zeitpunkt in einem *Gesamtzustand*, der als **Konfiguration** $(z, b, p) \in Z \times X^{\mathbb{Z}} \times \mathbb{Z}$ bezeichnet wird

Vollständig beschrieben durch...

Konfiguration einer TM

Eine Turingmaschine befindet sich zu jedem Zeitpunkt in einem *Gesamtzustand*, der als **Konfiguration** $(z, b, p) \in Z \times X^{\mathbb{Z}} \times \mathbb{Z}$ bezeichnet wird

Vollständig beschrieben durch...

- den aktuellen **Zustand** $z \in Z$ der Steuereinheit,

Konfiguration einer TM

Eine Turingmaschine befindet sich zu jedem Zeitpunkt in einem *Gesamtzustand*, der als **Konfiguration** $(z, b, p) \in Z \times X^{\mathbb{Z}} \times \mathbb{Z}$ bezeichnet wird

Vollständig beschrieben durch...

- den aktuellen **Zustand** $z \in Z$ der Steuereinheit,
- die aktuelle **Beschriftung des gesamten Bandes**, die man als Abbildung $b : \mathbb{Z} \to X$ formalisieren kann, und

Konfiguration einer TM

Eine Turingmaschine befindet sich zu jedem Zeitpunkt in einem *Gesamtzustand*, der als **Konfiguration** $(z, b, p) \in Z \times X^{\mathbb{Z}} \times \mathbb{Z}$ bezeichnet wird

Vollständig beschrieben durch...

- den aktuellen **Zustand** $z \in Z$ der Steuereinheit,
- die aktuelle **Beschriftung des gesamten Bandes**, die man als Abbildung $b : \mathbb{Z} \to X$ formalisieren kann, und
- die aktuelle **Position** $p \in Z$ **des Kopfes**.

Berechnungsschritt, d.h. Konfigurationswechsel

Berechnungsschritt, d.h. Konfigurationswechsel

Karlsruhe Institute of Technology

Die **Menge aller Konfigurationen** bezeichnen wir als $\mathbb{C}_t$

Berechnungsschritt, d.h. Konfigurationswechsel

Die **Menge aller Konfigurationen** bezeichnen wir als $\mathbb{C}_t$

Schritt einer TM

- $\Delta_x : \mathbb{C}_t \dashrightarrow \mathbb{C}_t$
- $\Delta_1(c)$ liefert direkte Nachfolgekonfiguration zu c

Berechnungsschritt, d.h. Konfigurationswechsel

Die **Menge aller Konfigurationen** bezeichnen wir als $\mathbb{C}_t$

Schritt einer TM

- $\Delta_x : \mathbb{C}_t \dashrightarrow \mathbb{C}_t$
- $\Delta_1(c)$ liefert direkte Nachfolgekonfiguration zu c

Endkonfigurationen einer TM

ist erreicht, falls $\Delta_1(c)$ nicht definiert ist

Arten der Berechnung

Arten der Berechnung

endliche Berechnung

- endliche Folge von Konfigurationen $(c_0, c_1, c_2, ..., c_t)$,
- wobei $0 < i \leq t$ gilt $c_i = \Delta_1(c_{i-1})$

Arten der Berechnung

endliche Berechnung

- endliche Folge von Konfigurationen $(c_0, c_1, c_2, ..., c_t)$,
- wobei $0 < i \leq t$ gilt $c_i = \Delta_1(c_{i-1})$

haltende Berechnung

- endliche Berechnung
- deren letzte Konfiguration eine Endkonfiguration ist

Arten der Berechnung

endliche Berechnung

- endliche Folge von Konfigurationen $(c_0, c_1, c_2, ..., c_t)$,
- wobei $0 < i \leq t$ gilt $c_i = \Delta_1(c_{i-1})$

haltende Berechnung

- endliche Berechnung
- deren letzte Konfiguration eine Endkonfiguration ist

unendliche Berechnung

- unendliche Folge von Konfigurationen $(c_0, c_1, c_2, ...)$
- wobei für $i > 0$ gilt $c_i = \Delta_1(c_{i-1})$
- **nicht haltend**

Turingmaschinenakzeptoren

Turingmaschinenakzeptoren

analog zu endlichen Automaten

- **Erkennung formaler Sprachen**: ein Bit akzeptiert/abgelehnt

Turingmaschinenakzeptoren

analog zu endlichen Automaten

- **Erkennung formaler Sprachen**: ein Bit akzeptiert/abgelehnt
- Teilmenge $F \subset Z$ **akzeptierender Zustände**
- TM akzeptiert Eingabewort w, wenn

Turingmaschinenakzeptoren

analog zu endlichen Automaten

- **Erkennung formaler Sprachen**: ein Bit akzeptiert/abgelehnt
- Teilmenge $F \subset Z$ **akzeptierender Zustände**
- TM akzeptiert Eingabewort w, wenn
  - TM für Eingabe w hält und

Turingmaschinenakzeptoren

Karlsruhe Institute of Technology

analog zu endlichen Automaten

- **Erkennung formaler Sprachen**: ein Bit akzeptiert/abgelehnt
- Teilmenge $F \subset Z$ **akzeptierender Zustände**
- TM akzeptiert Eingabewort w, wenn
  - TM für Eingabe w hält und
  - der Zustand der Endkonfiguration $\Delta_*(c_0(w))$ akzepierend ist

Turingmaschinenakzeptoren

analog zu endlichen Automaten

- **Erkennung formaler Sprachen**: ein Bit akzeptiert/abgelehnt
- Teilmenge $F \subset Z$ **akzeptierender Zustände**
- TM akzeptiert Eingabewort w, wenn
  - TM für Eingabe w hält und
  - der Zustand der Endkonfiguration $\Delta_*(c_0(w))$ akzepierend ist
- $L(T)$: Menge der akzeptierten Wörter

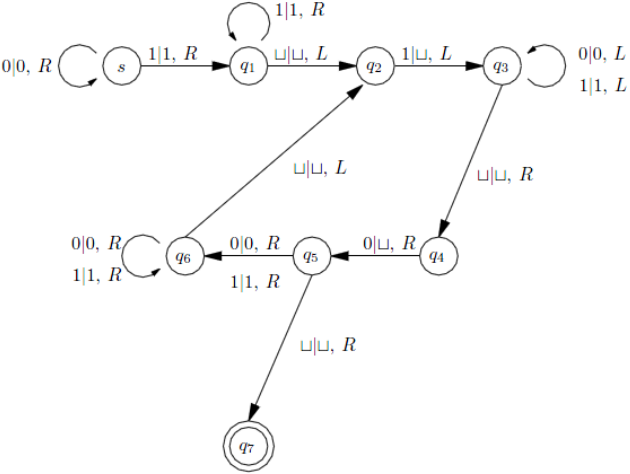
Ihr seid dran...

Aufgabe

Gebt ein TM-Akzeptor an, der die Sprache $L^{=} = \{0^n 1^n : n \geq 1\}$ akzeptiert

Ihr seid dran...

Lösung (die Markierung des Startzustands fehlt)



Aufzählbare und entscheidbare Sprachen

zwei Möglichkeiten, wenn w von TM nicht akzeptiert wird

Aufzählbare und entscheidbare Sprachen

zwei Möglichkeiten, wenn w von TM nicht akzeptiert wird

1. TM hält für Eingabe w, aber Endzustand nicht akzeptierend

Aufzählbare und entscheidbare Sprachen

zwei Möglichkeiten, wenn w von TM nicht akzeptiert wird

1. TM hält für Eingabe w, aber Endzustand nicht akzeptierend
2. TM hält für Eingabe w **nicht**

Aufzählbare und entscheidbare Sprachen

Karlsruhe Institute of Technology

zwei Möglichkeiten, wenn w von TM nicht akzeptiert wird

1. TM hält für Eingabe w, aber Endzustand nicht akzeptierend
2. TM hält für Eingabe w **nicht**

Was wissen wir über die Berechnung?

Aufzählbare und entscheidbare Sprachen

zwei Möglichkeiten, wenn w von TM nicht akzeptiert wird

1. TM hält für Eingabe w, aber Endzustand nicht akzeptierend
2. TM hält für Eingabe w **nicht**

Was wissen wir über die Berechnung?

1. TM ist fertig und lehnt die Eingabe ab

Aufzählbare und entscheidbare Sprachen

zwei Möglichkeiten, wenn w von TM nicht akzeptiert wird

1. TM hält für Eingabe w, aber Endzustand nicht akzeptierend
2. TM hält für Eingabe w **nicht**

Was wissen wir über die Berechnung?

1. TM ist fertig und lehnt die Eingabe ab
2. TM ist noch nicht fertig (*Ob TM irgendwann w noch akzeptiert oder ablehnt, ist unklar!*)

Aufzählbare und entscheidbare Sprachen

zwei Möglichkeiten, wenn w von TM nicht akzeptiert wird

1. TM hält für Eingabe w, aber Endzustand nicht akzeptierend
2. TM hält für Eingabe w **nicht**

Was wissen wir über die Berechnung?

1. TM ist fertig und lehnt die Eingabe ab
2. TM ist noch nicht fertig (*Ob TM irgendwann w noch akzeptiert oder ablehnt, ist unklar!*)

Wir halten in zwei Definitionen fest

1. L heißt **entscheidbare Sprache**, wenn es eine TM gibt, die **immer hält** und L akzeptiert.

Aufzählbare und entscheidbare Sprachen

Karlsruhe Institute of Technology

zwei Möglichkeiten, wenn w von TM nicht akzeptiert wird

1. TM hält für Eingabe w, aber Endzustand nicht akzeptierend
2. TM hält für Eingabe w **nicht**

Was wissen wir über die Berechnung?

1. TM ist fertig und lehnt die Eingabe ab
2. TM ist noch nicht fertig (*Ob TM irgendwann w noch akzeptiert oder ablehnt, ist unklar!*)

Wir halten in zwei Definitionen fest

1. L heißt **entscheidbare Sprache**, wenn es eine TM gibt, die **immer hält** und L akzeptiert.
2. L heißt **aufzählbare**(semi-entscheidbar) **Sprache**, wenn es eine TM gibt, die L akzeptiert

Übersicht

Codierungen von Turingmaschinen

Gödelisierung

Codierungen von Turingmaschinen

Karlsruhe Institute of Technology

Gödelisierung

- Ziel: Beschreibe jede Turingmaschine durch ein Wort

Codierungen von Turingmaschinen

Gödelisierung

- Ziel: Beschreibe jede Turingmaschine durch ein Wort
- Verfahren: **Durch-Nummerierung** von Turingmaschinen
  - Triviales Alphabet zur Codierung wie z.B. $A = \{[,], 0, 1\}$

Codierungen von Turingmaschinen

Gödelisierung

- Ziel: Beschreibe jede Turingmaschine durch ein Wort
- Verfahren: **Durch-Nummerierung** von Turingmaschinen
  - Triviales Alphabet zur Codierung wie z.B. $A = \{[,], 0, 1\}$
  - Codierung der Zustände, Symbole, Kopfbewegungen, Funktionen

Codierungen von Turingmaschinen

Gödelisierung

- Ziel: Beschreibe jede Turingmaschine durch ein Wort
- Verfahren: **Durch-Nummerierung** von Turingmaschinen
  - Triviales Alphabet zur Codierung wie z.B. $A = \{[,], 0, 1\}$
  - Codierung der Zustände, Symbole, Kopfbewegungen, Funktionen
  - Codierung der gesamten Turingmaschine als Konkatentation aller codierten Bestandteile

Eigenschaften dieser Codierungen

einfache Syntaxanalyse ist möglich

- TM konstruierbar, die für $w \in A^*$ feststellt, ob es die Codierung einer TM ist oder nicht.

Eigenschaften dieser Codierungen

Karlsruhe Institute of Technology

einfache Syntaxanalyse ist möglich

- TM konstruierbar, die für $w \in A^*$ feststellt, ob es die Codierung einer TM ist oder nicht.

universelle Turingmaschine U existiert

Eigenschaften dieser Codierungen

einfache Syntaxanalyse ist möglich

- TM konstruierbar, die für $w \in A^*$ feststellt, ob es die Codierung einer TM ist oder nicht.

universelle Turingmaschine U existiert

- erhält als Eingabe zwei Argumente als Wort $[w_1][w_2]$

Eigenschaften dieser Codierungen

Karlsruhe Institute of Technology

einfache Syntaxanalyse ist möglich

- TM konstruierbar, die für $w \in A^*$ feststellt, ob es die Codierung einer TM ist oder nicht.

universelle Turingmaschine U existiert

- erhält als Eingabe zwei Argumente als Wort $[w_1][w_2]$
- prüft, ob w_1 Codierung einer Turingmaschine T ist

Eigenschaften dieser Codierungen

Karlsruhe Institute of Technology

einfache Syntaxanalyse ist möglich

- TM konstruierbar, die für $w \in A^*$ feststellt, ob es die Codierung einer TM ist oder nicht.

universelle Turingmaschine U existiert

- erhält als Eingabe zwei Argumente als Wort $[w_1][w_2]$
- prüft, ob w_1 Codierung einer Turingmaschine T ist
- falls nein: hält mit NEIN

Eigenschaften dieser Codierungen

einfache Syntaxanalyse ist möglich

- TM konstruierbar, die für $w \in A^*$ feststellt, ob es die Codierung einer TM ist oder nicht.

universelle Turingmaschine U existiert

- erhält als Eingabe zwei Argumente als Wort $[w_1][w_2]$
- prüft, ob w_1 Codierung einer Turingmaschine T ist
- falls nein: hält mit NEIN
- falls ja:

Eigenschaften dieser Codierungen

einfache Syntaxanalyse ist möglich

- TM konstruierbar, die für $w \in A^*$ feststellt, ob es die Codierung einer TM ist oder nicht.

universelle Turingmaschine U existiert

- erhält als Eingabe zwei Argumente als Wort $[w_1][w_2]$
- prüft, ob w_1 Codierung einer Turingmaschine T ist
- falls nein: hält mit NEIN
- falls ja:
    - U **simuliert Schritt für Schritt die Arbeit**, die T für die Eingabe w_2 durchführen würde

Eigenschaften dieser Codierungen

Karlsruhe Institute of Technology

einfache Syntaxanalyse ist möglich

- TM konstruierbar, die für $w \in A^*$ feststellt, ob es die Codierung einer TM ist oder nicht.

universelle Turingmaschine U existiert

- erhält als Eingabe zwei Argumente als Wort $[w_1][w_2]$
- prüft, ob w_1 Codierung einer Turingmaschine T ist
- falls nein: hält mit NEIN
- falls ja:
  - U **simuliert Schritt für Schritt die Arbeit**, die T für die Eingabe w_2 durchführen würde
  - U liefert am Ende als Ergebnis, was T liefern würde (*FALLS T hält!*)

Das Halteproblem

Das Halteproblem ist die formale Sprache

$H = \{w \in A^* | \text{w ist eine TM-Codierung und } T_w(w) \text{ hält. }\}$

Das Halteproblem

Das Halteproblem ist die formale Sprache

$H = \{w \in A^* | \text{w ist eine TM-Codierung und } T_w(w) \text{ hält. }\}$

Satz

Das **Halteproblem ist unentscheidbar**, d. h. es gibt keine TM, die das Problem entscheidet.

Das Halteproblem

Das Halteproblem ist die formale Sprache

$H = \{w \in A^* | \text{w ist eine TM-Codierung und } T_w(w) \text{ hält. }\}$

Satz

Das **Halteproblem ist unentscheidbar**, d. h. es gibt keine TM, die das Problem entscheidet.

Anmerkung

Aber: Das Halteproblem ist aufzählbar(semi-entscheidbar). Man zeigt das mittels Univeral-TM.

Übersicht

Guten Morgen...

Aufgabenblatt 12

Aufgabenblatt 13

Probeklausur

WDH: Turingmaschine

Unentscheidbare Probleme

Äquivalenzrelationen

Halbordnungen

Ordnungen

Abschluss

Definition von Äquivalenzrelationen

Karlsruhe Institute of Technology

Vorraussetzungen

reflexiv	xRx
transitiv	Aus xRy und yRz folgt xRz
symmetrisch	Aus xRy folgt yRx

Definition von Äquivalenzrelationen

Vorraussetzungen

reflexiv	xRx
transitiv	Aus xRy und yRz folgt xRz
symmetrisch	Aus xRy folgt yRx

Gelten alle diese Eigenschaften für alle x, y, handelt es sich um eine **Äquivalenzrelation**.

Äquivalenzrelationen

Karlsruhe Institute of Technology

Relation als Graph

- Darstellung von Relationen als gerichtete Graphen: Woran sieht man
  - Reflexivität?
  - Symmetrie?
  - Transitivität?

Äquivalenzrelationen

Relation als Graph

- Darstellung von Relationen als gerichtete Graphen: Woran sieht man
  - Reflexivität?
  - Symmetrie?
  - Transitivität?
- Wie sieht der Graph einer Äquivalenzrelation aus:

Äquivalenzrelationen

Karlsruhe Institute of Technology

Relation als Graph

- Darstellung von Relationen als gerichtete Graphen: Woran sieht man
  - Reflexivität?
  - Symmetrie?
  - Transitivität?
- Wie sieht der Graph einer Äquivalenzrelation aus: „Cliquen“, in denen jeder mit jedem verbunden ist
- dazwischen nichts (die Cliquen heißen später Äquivalenzklassen)

Äquivalenzrelationen von Nerode

Definition

für alle $w_1, w_2 \in A^*$ ist

$w1 \equiv_L w2 \Leftrightarrow \left(\forall w \in A^* : w_1 w \in L \Leftrightarrow w_2 w \in L\right)$

Äquivalenzrelationen von Nerode

Definition

für alle $w_1, w_2 \in A^*$ ist

$w1 \equiv_L w2 \Leftrightarrow (\forall w \in A^* : w_1 w \in L \Leftrightarrow w_2 w \in L)$

- das liest man besser mehrmals durch

Äquivalenzrelationen von Nerode

Definition
für alle $w_1, w_2 \in A^*$ ist
$w1 \equiv_L w2 \Leftrightarrow (\forall w \in A^* : w_1 w \in L \Leftrightarrow w_2 w \in L)$

- das liest man besser mehrmals durch
  - man nehme eine Sprache L, die von einem endlichen Akzeptor erkannt wird
  - man nehme zwei Wörter w_1, w_2 die *nicht* $\equiv_L$-äquivalent sind
  - Was kann man über $f^*(z_0, w_1)$ und $f^*(z_0, w_2)$ sagen?

Äquivalenzrelationen von Nerode

Definition
für alle $w_1, w_2 \in A^*$ ist
$w1 \equiv_L w2 \Leftrightarrow (\forall w \in A^* : w_1 w \in L \Leftrightarrow w_2 w \in L)$

- das liest man besser mehrmals durch
  - man nehme eine Sprache L, die von einem endlichen Akzeptor erkannt wird
  - man nehme zwei Wörter w_1, w_2 die *nicht* $\equiv_L$-äquivalent sind
  - Was kann man über $f^*(z_0, w_1)$ und $f^*(z_0, w_2)$ sagen?
  - Sie müssen verschieden sein, denn sonst $f^*(z_0, w_1) = f^*(z_0, w_2)$ und dann auch für jedes Suffix w: $f^*(z_0, w_1 w) = f^*(z_0, w_2 w)$, also werden für jedes Suffix entweder beide Wörter $w_1 w$ und $w_2 w$ oder keines akzeptiert, und dann wären w_1 und w_2 ja äquivalent.

Äquivalenzrelationen von Nerode

Beispiele

aus dem Skript:
Sei $L = \langle a * b* \rangle \subset A^*$

- $w_1 = aaa, w_2 = a$
- $w_1 = aaab, w_2 = abb$
- $w_1 = aa, w_2 = abb$
- $w_1 = aba, w_2 = babb$
- $w_1 = ab, w_2 = ba$

Äquivalenzrelationen von Nerode

Beispiele

aus dem Skript:
Sei $L = \langle a * b * \rangle \subset A^*$

- $w_1 = aaa, w_2 = a$
- $w_1 = aaab, w_2 = abb$
- $w_1 = aa, w_2 = abb$ nicht $\equiv_L$-äquivalent sind
- $w_1 = aba, w_2 = babb$
- $w_1 = ab, w_2 = ba$ nicht $\equiv_L$-äquivalent sind

Übersicht

Guten Morgen...

Aufgabenblatt 12

Aufgabenblatt 13

Probeklausur

WDH: Turingmaschine

Unentscheidbare Probleme

Äquivalenzrelationen

Halbordnungen

Ordnungen

Abschluss

WDH: Definition von Äquivalenzrelationen

Vorraussetzungen

WDH: Definition von Äquivalenzrelationen

Vorraussetzungen

reflexiv	xRx
transitiv	Aus xRy und yRz folgt xRz
symmetrisch	Aus xRy folgt yRx

WDH: Definition von Äquivalenzrelationen

Karlsruhe Institute of Technology

Vorraussetzungen

reflexiv	xRx
transitiv	Aus xRy und yRz folgt xRz
symmetrisch	Aus xRy folgt yRx

Gelten alle diese Eigenschaften für alle $x, y, z \in M$, handelt es sich um eine **Äquivalenzrelation**.

Definition einer Halbordnung

Vorraussetzungen

reflexiv	xRx
transitiv	Aus xRy und yRz folgt xRz

Definition einer Halbordnung

Vorraussetzungen

reflexiv	xRx
transitiv	Aus xRy und yRz folgt xRz
antisymmetrisch	Aus xRy und yRx folgt $x = y$

Definition einer Halbordnung

Vorraussetzungen

reflexiv	xRx
transitiv	Aus xRy und yRz folgt xRz
antisymmetrisch	Aus xRy und yRx folgt $x = y$

- Gelten alle diese Eigenschaften für alle x, y, handelt es sich bei $R \subseteq MxM$ um eine **Halbordnung**.

Definition einer Halbordnung

Vorraussetzungen

reflexiv	xRx
transitiv	Aus xRy und yRz folgt xRz
antisymmetrisch	Aus xRy und yRx folgt $x = y$

- Gelten alle diese Eigenschaften für alle x, y, handelt es sich bei $R \subseteq MxM$ um eine **Halbordnung**.
- Wenn R Halbordnung auf Menge M ist, nennt man M eine **halbgeordnete Menge**.

Beispiel Mengeninklusion

Untersuchung der Mengeninklusion

Handelt es sich bei der Relation $\subseteq$ (Mengeninklusion) um eine **Äquivalenzrelation oder Halbordnung** auf Potenzmenge $P = 2^M$?

Beispiel Mengeninklusion

Untersuchung der Mengeninklusion

Handelt es sich bei der Relation $\subseteq$ (Mengeninklusion) um eine **Äquivalenzrelation oder Halbordnung** auf Potenzmenge $P = 2^M$?

- relfexiv: $\forall A \in P$: $A \subseteq A$

Beispiel Mengeninklusion

Karlsruhe Institute of Technology

Untersuchung der Mengeninklusion

Handelt es sich bei der Relation $\subseteq$ (Mengeninklusion) um eine **Äquivalenzrelation oder Halbordnung** auf Potenzmenge $P = 2^M$?

- relfexiv: $\forall A \in P\colon A \subseteq A$
- transitiv: $\forall A, B, C \in P\colon A \subseteq B$ und $B \subseteq C \Longrightarrow A \subseteq C$

Beispiel Mengeninklusion

Untersuchung der Mengeninklusion

Handelt es sich bei der Relation $\subseteq$ (Mengeninklusion) um eine **Äquivalenzrelation oder Halbordnung** auf Potenzmenge $P = 2^M$?

- relfexiv: $\forall A \in P\colon A \subseteq A$
- transitiv: $\forall A, B, C \in P\colon A \subseteq B$ und $B \subseteq C \Longrightarrow A \subseteq C$
- symmetrisch: $\forall A, B \in P\colon A \subseteq B \Longrightarrow B \subseteq A$

Beispiel Mengeninklusion

Untersuchung der Mengeninklusion

Handelt es sich bei der Relation $\subseteq$ (Mengeninklusion) um eine **Äquivalenzrelation oder Halbordnung** auf Potenzmenge $P = 2^M$?

- relfexiv: $\forall A \in P$: $A \subseteq A$
- transitiv: $\forall A, B, C \in P$: $A \subseteq B$ und $B \subseteq C \Longrightarrow A \subseteq C$
- symmetrisch: $\forall A, B \in P$: $A \subseteq B \Longrightarrow B \subseteq A$ gilt nicht. **ABER:** Aus **keiner Symmetrie** folgt nicht notwendig die Antisymmetrie!

Beispiel Mengeninklusion

Untersuchung der Mengeninklusion

Handelt es sich bei der Relation $\subseteq$ (Mengeninklusion) um eine **Äquivalenzrelation oder Halbordnung** auf Potenzmenge $P = 2^M$?

- relfexiv: $\forall A \in P$: $A \subseteq A$
- transitiv: $\forall A, B, C \in P$: $A \subseteq B$ und $B \subseteq C \Longrightarrow A \subseteq C$
- symmetrisch: $\forall A, B \in P$: $A \subseteq B \Longrightarrow B \subseteq A$ gilt nicht. **ABER:** Aus **keiner Symmetrie** folgt nicht notwendig die Antisymmetrie!
- antisymmetrisch: $\forall A, B \in P$: $A \subseteq B$ und $B \subseteq A \Longrightarrow A = B$ (Analogie zur Mengengleichheit)

Beispiel Mengeninklusion

Karlsruhe Institute of Technology

Untersuchung der Mengeninklusion

Handelt es sich bei der Relation $\subseteq$ (Mengeninklusion) um eine **Äquivalenzrelation oder Halbordnung** auf Potenzmenge $P = 2^M$?

- relfexiv: $\forall A \in P\colon A \subseteq A$
- transitiv: $\forall A, B, C \in P\colon A \subseteq B$ und $B \subseteq C \Longrightarrow A \subseteq C$
- symmetrisch: $\forall A, B \in P\colon A \subseteq B \Longrightarrow B \subseteq A$ gilt nicht. **ABER:** Aus **keiner Symmetrie** folgt nicht notwendig die Antisymmetrie!
- antisymmetrisch: $\forall A, B \in P\colon A \subseteq B$ und $B \subseteq A \Longrightarrow A = B$ (Analogie zur Mengengleichheit)

Die Mengeninklusion ist eine Halbordnung.

Ihr seid dran...

Aufgabe

Überprüft, ob es sich bei folgenden Relationen um Halbordnungen handelt:

- $\sqsubseteq_p$ auf A^* mit $v \sqsubseteq_p w \Leftrightarrow \exists u : vu = w$?

Ihr seid dran...

Aufgabe

Überprüft, ob es sich bei folgenden Relationen um Halbordnungen handelt:

- $\sqsubseteq_p$ auf A^* mit $v \sqsubseteq_p w \Leftrightarrow \exists u : vu = w$?
  - **Reflexivität**: gilt wegen $w_1\epsilon = w_1$
  - **Antisymmetrie**: wenn $w_1 \sqsubseteq_p w_2$ und $w_2 \sqsubseteq w_1$, dann gibt es $u_1, u_2 \in A^*$ mit $w_1 u_1 = w_2$ und $w_2 u_2 = w_1$. Also ist $w_1 u_1 u_2 = w_2 u_2 = w_1$. Also muss $|u_1 u_2| = 0$ sein, also $u_1 = u_2 = \epsilon$, also $w_1 = w_2$.
  - **Transitivität**: wenn $w_1 \sqsubseteq_p w_2$ und $w_2 \sqsubseteq w_3$, dann gibt es $u_1, u_2 \in A^*$ mit $w_1 u_1 = w_2$ und $w_2 u_2 = w_3$. Also ist $w_1(u_1 u_2) = (w_1 u_1)u_2 = w_2 u_2 = w_3$, also $w_1 \sqsubseteq w_3$.

Ihr seid dran...

Aufgabe

Überprüft, ob es sich bei folgenden Relationen um Halbordnungen handelt:

- $\sqsubseteq_p$ auf A^* mit $v \sqsubseteq_p w \Leftrightarrow \exists u : vu = w$?
    - **Reflexivität**: gilt wegen $w_1 \epsilon = w_1$
    - **Antisymmetrie**: wenn $w_1 \sqsubseteq_p w_2$ und $w_2 \sqsubseteq w_1$, dann gibt es $u_1, u_2 \in A^*$ mit $w_1 u_1 = w_2$ und $w_2 u_2 = w_1$. Also ist $w_1 u_1 u_2 = w_2 u_2 = w_1$. Also muss $|u_1 u_2| = 0$ sein, also $u_1 = u_2 = \epsilon$, also $w_1 = w_2$.
    - **Transitivität**: wenn $w_1 \sqsubseteq_p w_2$ und $w_2 \sqsubseteq w_3$, dann gibt es $u_1, u_2 \in A^*$ mit $w_1 u_1 = w_2$ und $w_2 u_2 = w_3$. Also ist $w_1(u_1 u_2) = (w_1 u_1) u_2 = w_2 u_2 = w_3$, also $w_1 \sqsubseteq w_3$.
- $\sqsubseteq$ auf A^* mit $w_1 \sqsubseteq w_2 \Leftrightarrow |w_1| \leq |w_2|$?

Ihr seid dran...

Aufgabe

Überprüft, ob es sich bei folgenden Relationen um Halbordnungen handelt:

- $\sqsubseteq_p$ auf A^* mit $v \sqsubseteq_p w \Leftrightarrow \exists u : vu = w$?
  - **Reflexivität**: gilt wegen $w_1\epsilon = w_1$
  - **Antisymmetrie**: wenn $w_1 \sqsubseteq_p w_2$ und $w_2 \sqsubseteq w_1$, dann gibt es $u_1, u_2 \in A^*$ mit $w_1 u_1 = w_2$ und $w_2 u_2 = w_1$. Also ist $w_1 u_1 u_2 = w_2 u_2 = w_1$. Also muss $|u_1 u_2| = 0$ sein, also $u_1 = u_2 = \epsilon$, also $w_1 = w_2$.
  - **Transitivität**: wenn $w_1 \sqsubseteq_p w_2$ und $w_2 \sqsubseteq w_3$, dann gibt es $u_1, u_2 \in A^*$ mit $w_1 u_1 = w_2$ und $w_2 u_2 = w_3$. Also ist $w_1(u_1 u_2) = (w_1 u_1)u_2 = w_2 u_2 = w_3$, also $w_1 \sqsubseteq w_3$.
- $\sqsubseteq$ auf A^* mit $w_1 \sqsubseteq w_2 \Leftrightarrow |w_1| \leq |w_2|$?
  - Antisymmetrie ist verletzt.

Ihr seid dran...

Aufgabe

Überprüft, ob es sich bei folgenden Relationen um Halbordnungen handelt:

- $\sqsubseteq_p$ auf A^* mit $v \sqsubseteq_p w \Leftrightarrow \exists u : vu = w$?
  - **Reflexivität**: gilt wegen $w_1\epsilon = w_1$
  - **Antisymmetrie**: wenn $w_1 \sqsubseteq_p w_2$ und $w_2 \sqsubseteq w_1$, dann gibt es $u_1, u_2 \in A^*$ mit $w_1 u_1 = w_2$ und $w_2 u_2 = w_1$. Also ist $w_1 u_1 u_2 = w_2 u_2 = w_1$. Also muss $|u_1 u_2| = 0$ sein, also $u_1 = u_2 = \epsilon$, also $w_1 = w_2$.
  - **Transitivität**: wenn $w_1 \sqsubseteq_p w_2$ und $w_2 \sqsubseteq w_3$, dann gibt es $u_1, u_2 \in A^*$ mit $w_1 u_1 = w_2$ und $w_2 u_2 = w_3$. Also ist $w_1(u_1 u_2) = (w_1 u_1)u_2 = w_2 u_2 = w_3$, also $w_1 \sqsubseteq w_3$.
- $\sqsubseteq$ auf A^* mit $w_1 \sqsubseteq w_2 \Leftrightarrow |w_1| \leq |w_2|$?
  - Antisymmetrie ist verletzt.
  - Reflexivität und Transitivität sind erfüllt.

Hassediagramm

Konstruktion

Zur **Veranschaulichung einer Halbordnung** lassen sich Hassediagramme folgendermaßen erstellen:

Hassediagramm

Konstruktion

Zur **Veranschaulichung einer Halbordnung** lassen sich Hassediagramme folgendermaßen erstellen:

1. Darstellung der Halbordnung als Graph

Hassediagramm

Konstruktion

Zur **Veranschaulichung einer Halbordnung** lassen sich Hassediagramme folgendermaßen erstellen:

1. Darstellung der Halbordnung als Graph
2. Entfernen aller reflexiven und transitiven Kanten

Extreme „Elemente“ I

Sei $(M, \sqsubseteq)$ halbgeordnet und $T \subseteq M$

Extreme „Elemente“ I

Sei $(M, \sqsubseteq)$ halbgeordnet und $T \subseteq M$

minimale und maximale Elemente

- $x \in T$ heißt **minimales Element** von T, wenn es kein $y \in T$ gibt mit $y \sqsubseteq x$ und $y \neq x$.

Extreme „Elemente“ I

Sei $(M, \sqsubseteq)$ halbgeordnet und $T \subseteq M$

minimale und maximale Elemente

- $x \in T$ heißt **minimales Element** von T, wenn es kein $y \in T$ gibt mit $y \sqsubseteq x$ und $y \neq x$.
- $x \in T$ heißt **maximales Element** von T, wenn es kein $y \in T$ gibt mit $x \sqsubseteq y$ und $x \neq y$.

Extreme „Elemente" I

Sei $(M, \sqsubseteq)$ halbgeordnet und $T \subseteq M$

minimale und maximale Elemente

- $x \in T$ heißt **minimales Element** von T, wenn es kein $y \in T$ gibt mit $y \sqsubseteq x$ und $y \neq x$.
- $x \in T$ heißt **maximales Element** von T, wenn es kein $y \in T$ gibt mit $x \sqsubseteq y$ und $x \neq y$.

kleinstes und größtes Element

- $x \in T$ heißt **kleinstes Element** von T, wenn für alle $y \in T$ gilt: $x \sqsubseteq y$.

Extreme „Elemente“ I

Sei $(M, \sqsubseteq)$ halbgeordnet und $T \subseteq M$

minimale und maximale Elemente

- $x \in T$ heißt **minimales Element** von T, wenn es kein $y \in T$ gibt mit $y \sqsubseteq x$ und $y \neq x$.
- $x \in T$ heißt **maximales Element** von T, wenn es kein $y \in T$ gibt mit $x \sqsubseteq y$ und $x \neq y$.

kleinstes und größtes Element

- $x \in T$ heißt **kleinstes Element** von T, wenn für alle $y \in T$ gilt: $x \sqsubseteq y$.
- $x \in T$ heißt **größtes Element** von T, wenn für alle $y \in T$ gilt: $y \sqsubseteq x$.

Extreme „Elemente“ I

Sei $(M, \sqsubseteq)$ halbgeordnet und $T \subseteq M$

minimale und maximale Elemente

- $x \in T$ heißt **minimales Element** von T, wenn es kein $y \in T$ gibt mit $y \sqsubseteq x$ und $y \neq x$.
- $x \in T$ heißt **maximales Element** von T, wenn es kein $y \in T$ gibt mit $x \sqsubseteq y$ und $x \neq y$.

kleinstes und größtes Element

- $x \in T$ heißt **kleinstes Element** von T, wenn für alle $y \in T$ gilt: $x \sqsubseteq y$.
- $x \in T$ heißt **größtes Element** von T, wenn für alle $y \in T$ gilt: $y \sqsubseteq x$.

Eine Teilmenge T kann mehrere minimale (bzw. maximale) Elemente besitzen, aber nur ein kleinstes (bzw. größtes)!

Beispiel mit Hassediagramm

Karlsruhe Institute of Technology

Beispiel

- Male das Hassediagramm zur Halbordnung $(\{\{\}, a, b, c, ab, bc, ac\}, \subseteq)$

Beispiel mit Hassediagramm

Beispiel

- Male das Hassediagramm zur Halbordnung $(\{\{\}, a, b, c, ab, bc, ac\}, \subseteq)$
- woran erkennt man Minima?

Beispiel mit Hassediagramm

Beispiel

- Male das Hassediagramm zur Halbordnung $(\{\{\}, a, b, c, ab, bc, ac\}, \subseteq)$
- woran erkennt man Minima?
- woran Maxima?

Extreme „Elemente“ II

Sei $(M, \sqsubseteq)$ halbgeordnet und $T \subseteq M$

Extreme „Elemente“ II

Sei $(M, \sqsubseteq)$ halbgeordnet und $T \subseteq M$

Untere und obere Schranken

- $x \in M$ heißt **untere Schranke** von T, wenn für alle $y \in T$ gilt: $x \sqsubseteq y$.

Extreme „Elemente“ II

Sei $(M, \sqsubseteq)$ halbgeordnet und $T \subseteq M$

Untere und obere Schranken

- $x \in M$ heißt **untere Schranke** von T, wenn für alle $y \in T$ gilt: $x \sqsubseteq y$.
- $x \in M$ heißt **obere Schranke** von T, wenn für alle $y \in T$ gilt: $y \sqsubseteq x$.

Extreme „Elemente“ II

Sei $(M, \sqsubseteq)$ halbgeordnet und $T \subseteq M$

Untere und obere Schranken

- $x \in M$ heißt **untere Schranke** von T, wenn für alle $y \in T$ gilt: $x \sqsubseteq y$.
- $x \in M$ heißt **obere Schranke** von T, wenn für alle $y \in T$ gilt: $y \sqsubseteq x$.

Also: Schranken von T dürfen außerhalb von T liegen.

Extreme „Elemente“ III

Supremum und Infimum

- Besitzt die Menge aller oberen Schranken einer Teilmenge T ein kleinstes Element, so heißt dies das **Supremum** von T ($sup(T)$)

Extreme „Elemente“ III

Supremum und Infimum

- Besitzt die Menge aller oberen Schranken einer Teilmenge T ein kleinstes Element, so heißt dies das **Supremum** von T ($sup(T)$)
- Besitzt die Menge aller unteren Schranken einer Teilmenge T ein größtes Element, so heißt dies das **Infimum** von T ($inf(T)$)

Extreme „Elemente“ III

Karlsruhe Institute of Technology

Supremum und Infimum

- Besitzt die Menge aller oberen Schranken einer Teilmenge T ein kleinstes Element, so heißt dies das **Supremum** von T ($sup(T)$)
- Besitzt die Menge aller unteren Schranken einer Teilmenge T ein größtes Element, so heißt dies das **Infimum** von T ($inf(T)$)
- **Achtung**: Existieren nicht, wenn
    - überhaupt keine oberen (unteren) Schranken vorhanden

Extreme „Elemente" III

Supremum und Infimum

- Besitzt die Menge aller oberen Schranken einer Teilmenge T ein kleinstes Element, so heißt dies das **Supremum** von T ($sup(T)$)
- Besitzt die Menge aller unteren Schranken einer Teilmenge T ein größtes Element, so heißt dies das **Infimum** von T ($inf(T)$)
- **Achtung**: Existieren nicht, wenn
  - überhaupt keine oberen (unteren) Schranken vorhanden
  - keine eindeutig kleinste (größte) Schranke aller oberer (unterer) Schranken

Vollständige Halbordnungen

aufsteigende Kette

wird definiert als

- abzählbar unendliche Folge $(x_0, x_1, x_2, \ldots)$ von Elementen
- mit Eigenschaft: $\forall i \in N_0 \colon x_i \sqsubseteq x_{i+1}$

Vollständige Halbordnungen

aufsteigende Kette

wird definiert als

- abzählbar unendliche Folge $(x_0, x_1, x_2, ...)$ von Elementen
- mit Eigenschaft: $\forall i \in N_0 \colon x_i \sqsubseteq x_{i+1}$

vollständige Halbordnung

Eine Halbordnung heißt **vollständig**, wenn

- sie ein kleinstes Element $\perp$ hat und
- jede aufsteigende Kette $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq ...$ ein Supremum x_i besitzt

Vollständige Halbordnungen II

Stetige Abbildungen auf vollständigen Halbordnungen

- Beispiel aus dem Skript
- Gegeben sei:
- Terminalzeichenalphabet $T = \{a, b\}$,

Vollständige Halbordnungen II

Stetige Abbildungen auf vollständigen Halbordnungen

- Beispiel aus dem Skript
- Gegeben sei:
- Terminalzeichenalphabet $T = \{a, b\}$,
- D die halbgeordnete Potenzmenge $D = 2^{T^*}$ der Menge aller Wörter
- mit Inklusion als Halbordnungsrelation.

Vollständige Halbordnungen II

Stetige Abbildungen auf vollständigen Halbordnungen

- Beispiel aus dem Skript
- Gegeben sei:
- Terminalzeichenalphabet $T = \{a, b\}$,
- D die halbgeordnete Potenzmenge $D = 2^{T^*}$ der Menge aller Wörter
- mit Inklusion als Halbordnungsrelation.
- Elemente der Halbordnung sind also Mengen von Wörtern, d.h. formale Sprachen.

Vollständige Halbordnungen II

Stetige Abbildungen auf vollständigen Halbordnungen

- Beispiel aus dem Skript
- Gegeben sei:
- Terminalzeichenalphabet $T = \{a, b\}$,
- D die halbgeordnete Potenzmenge $D = 2^{T^*}$ der Menge aller Wörter
- mit Inklusion als Halbordnungsrelation.
- Elemente der Halbordnung sind also Mengen von Wörtern, d.h. formale Sprachen.
- Kleinstes Element der Halbordnung ist die leere Menge $\emptyset$.

Vollständige Halbordnungen II

Karlsruhe Institute of Technology

Stetige Abbildungen auf vollständigen Halbordnungen

- Beispiel aus dem Skript
- Gegeben sei:
- Terminalzeichenalphabet $T = \{a, b\}$,
- D die halbgeordnete Potenzmenge $D = 2^{T^*}$ der Menge aller Wörter
- mit Inklusion als Halbordnungsrelation.
- Elemente der Halbordnung sind also Mengen von Wörtern, d.h. formale Sprachen.
- Kleinstes Element der Halbordnung ist die leere Menge $\emptyset$.
- Wie weiter vorne erwähnt, ist diese Halbordnung vollständig.

Vollständige Halbordnungen III

Beweis

- Es sei $v \in T^*$ ein Wort und $f_v : D \to D$ die Abbildung $f_v(L) = \{v\}L$, die vor jedes Wort von L vorne v konkateniert.
- Behauptung: f_v ist stetig.
- Beweis: Es sei $L_0 \subseteq L_1 \subseteq L_2 \subseteq \cdots$ eine Kette und $L = \bigcup L_i$ ihr Supremum.
  $f_v(L_i) = \{vw | w \in L_i\}$, also
  $\bigcup_i f_v(L_i) = \{vw | \exists i \in N_0 : w \in L_i\} = \{v\}\{w | \exists i \in N_0 : w \in L_i\}$
  $= \{v\} \bigcup_i L_i = f(\bigcup_i L_i)$.
- analog für Konkatenation von rechts

Übersicht

Totale Ordnung

Definition

Relation $R \subseteq MxM$ ist eine Ordnung oder genauer **totale Ordnung**, wenn

- R Halbordnung ist

Totale Ordnung

Definition

Relation $R \subseteq MxM$ ist eine Ordnung oder genauer **totale Ordnung**, wenn

- R Halbordnung ist
- und gilt: $\forall x, y \in M$: $xRy \vee yRx$

Totale Ordnung

Definition

Relation $R \subseteq MxM$ ist eine Ordnung oder genauer **totale Ordnung**, wenn

- R Halbordnung ist
- und gilt: $\forall x, y \in M$: $xRy \vee yRx$

Anmerkungen

- $\Rightarrow$: Es gibt keine unvergleichbaren Elemente.

Totale Ordnung

Beispiele

- $(N_0, \leq)$

Totale Ordnung

Beispiele

- $(N_0, \leq)$
- $(\{a, b\}^*, \sqsubseteq_1)$ mit $w_1 \sqsubseteq_1 w_2$ „wie im Wörterbuch“

Totale Ordnung

Beispiele

- $(N_0, \leq)$
- $(\{a, b\}^*, \sqsubseteq_1)$ mit $w_1 \sqsubseteq_1 w_2$ „wie im Wörterbuch“
- $(\{a, b\}^*, \sqsubseteq_2)$ mit $w_1 \sqsubseteq_2 w_2$ genau dann, wenn

Totale Ordnung

Beispiele

- $(N_0, \leq)$
- $(\{a, b\}^*, \sqsubseteq_1)$ mit $w_1 \sqsubseteq_1 w_2$ „wie im Wörterbuch“
- $(\{a, b\}^*, \sqsubseteq_2)$ mit $w_1 \sqsubseteq_2 w_2$ genau dann, wenn
  - entweder $|w_1| < |w_2|$

Totale Ordnung

Beispiele

- $(N_0, \leq)$
- $(\{a, b\}^*, \sqsubseteq_1)$ mit $w_1 \sqsubseteq_1 w_2$ „wie im Wörterbuch“
- $(\{a, b\}^*, \sqsubseteq_2)$ mit $w_1 \sqsubseteq_2 w_2$ genau dann, wenn
  - entweder $|w_1| < |w_2|$
  - oder $|w_1| = |w_2|$ und $w_1 \sqsubseteq_1 w_2$ gilt

Ihr seid dran...

Beispiele für $\sqsubseteq_1$:

- Warum ist $aa \sqsubseteq_1 aabba$?

Ihr seid dran...

Beispiele für $\sqsubseteq_1$:

- Warum ist $aa \sqsubseteq_1 aabba$?
- Warum ist $aa \sqsubseteq_1 bba$?

Ihr seid dran...

Beispiele für $\sqsubseteq_1$:

- Warum ist $aa \sqsubseteq_1 aabba$?
- Warum ist $aa \sqsubseteq_1 bba$?
- Warum ist $aaaaa \sqsubseteq_1 bba$?

Ihr seid dran...

Beispiele für $\sqsubseteq_1$:

- Warum ist $aa \sqsubseteq_1 aabba$?
- Warum ist $aa \sqsubseteq_1 bba$?
- Warum ist $aaaaa \sqsubseteq_1 bba$?
- Warum ist $aaaab \sqsubseteq_1 aab$?

Ihr seid dran...

Beispiele für $\sqsubseteq_1$:

- Warum ist $aa \sqsubseteq_1 aabba$?
- Warum ist $aa \sqsubseteq_1 bba$?
- Warum ist $aaaaa \sqsubseteq_1 bba$?
- Warum ist $aaaab \sqsubseteq_1 aab$?

Beispiele für $\sqsubseteq_2$:

- Warum ist $aa \sqsubseteq_2 aabba$?

Ihr seid dran...

Beispiele für $\sqsubseteq_1$:

- Warum ist $aa \sqsubseteq_1 aabba$?
- Warum ist $aa \sqsubseteq_1 bba$?
- Warum ist $aaaaa \sqsubseteq_1 bba$?
- Warum ist $aaaab \sqsubseteq_1 aab$?

Beispiele für $\sqsubseteq_2$:

- Warum ist $aa \sqsubseteq_2 aabba$?
- Warum ist $aa \sqsubseteq_2 bba$?

Ihr seid dran...

Beispiele für $\sqsubseteq_1$:

- Warum ist $aa \sqsubseteq_1 aabba$?
- Warum ist $aa \sqsubseteq_1 bba$?
- Warum ist $aaaaa \sqsubseteq_1 bba$?
- Warum ist $aaaab \sqsubseteq_1 aab$?

Beispiele für $\sqsubseteq_2$:

- Warum ist $aa \sqsubseteq_2 aabba$?
- Warum ist $aa \sqsubseteq_2 bba$?
- Warum ist $bba \sqsubseteq_2 aaaaa$? (vergleiche $\sqsubseteq_1$!)

Ihr seid dran...

Beispiele für $\sqsubseteq_1$:

- Warum ist $aa \sqsubseteq_1 aabba$?
- Warum ist $aa \sqsubseteq_1 bba$?
- Warum ist $aaaaa \sqsubseteq_1 bba$?
- Warum ist $aaaab \sqsubseteq_1 aab$?

Beispiele für $\sqsubseteq_2$:

- Warum ist $aa \sqsubseteq_2 aabba$?
- Warum ist $aa \sqsubseteq_2 bba$?
- Warum ist $bba \sqsubseteq_2 aaaaa$? (vergleiche $\sqsubseteq_1$!)
- Warum ist $aab \sqsubseteq_2 aaaab$? (vergleiche $\sqsubseteq_1$!)

Bleibt dran...

Aufgabe

Relation $\sqsubseteq_p$ auf $\{a, b\}^*$ eine totale Ordnung?

Bleibt dran...

Aufgabe

Relation $\sqsubseteq_p$ auf $\{a, b\}^*$ eine totale Ordnung?

Lösung

Es handelt sich um eine Halbordnung, allerdings mit unvergleichbaren Element wie z.B. a, b. Daher ist die Relation $\sqsubseteq_p$ **keine** totale Ordnung.

Übersicht

Zum Schluss...

Was ihr nun wissen solltet!

Zum Schluss...

Was ihr nun wissen solltet!

- Was ist ein Algorithmus?

Zum Schluss...

Was ihr nun wissen solltet!

- Was ist ein Algorithmus?
- Welche Arten von Berechnung unterscheiden wir?

Zum Schluss...

Was ihr nun wissen solltet!

- Was ist ein Algorithmus?
- Welche Arten von Berechnung unterscheiden wir?
- Was zeichnet das Halteproblem aus? Gibt es noch andere Probleme, auf die dasselbe zutrifft?

Zum Schluss...

Was ihr nun wissen solltet!

- Was ist ein Algorithmus?
- Welche Arten von Berechnung unterscheiden wir?
- Was zeichnet das Halteproblem aus? Gibt es noch andere Probleme, auf die dasselbe zutrifft?
- Was besagt die Äquivalenzrelation von Nerode?

Zum Schluss...

Was ihr nun wissen solltet!

- Was ist ein Algorithmus?
- Welche Arten von Berechnung unterscheiden wir?
- Was zeichnet das Halteproblem aus? Gibt es noch andere Probleme, auf die dasselbe zutrifft?
- Was besagt die Äquivalenzrelation von Nerode?

Ihr wisst was nicht?

Stellt **jetzt** Fragen!

Ende


WAS IST NUN MIT MEINEN STECHENDEN HERZSCHMERZEN, HERR DOKTOR?
DAS IST LIEBE. SIE SIND VERLIEBT. WOMÖGLICH IN MICH.
WWW.NICHTLUSTIG.DE