Karlsruhe Institute of Technology

Grundbegriffe der Informatik - Tutorium 21

Christian Jülg

Wintersemester 2012/13

22. Januar 2013

http://gbi-tutor.blogspot.com

Übersicht

Karlsruhe Institute of Technology

Guten Morgen...

Aufgabenblatt 11

Aufgabenblatt 12

Reguläre Ausdrücke

Rechtslineare Grammatiken

Kantorowitsch-Bäume, Regex-Bäume

Berechenbarkeit

Turingmaschinen

Abschluss

Übersicht

Karlsruhe Institute of Technology

Zum Warmwerden...

Der Reguläre Ausdruck $ab*c*a$ ...

1. ... beschreibt eine Typ-3 Sprache.
2. ... beschreibt die Sprache $L_1 = \{ab^n c^n a | n \in \mathbb{N}_0\}$
3. ... beschreibt eine endliche Sprache.

Rechtslineare Grammatiken ...

1. ... beschreiben reguläre Sprachen.
2. ... dürfen ϵ-Produktionen enthalten.
3. ... lassen sich in einen endlichen Automaten überführen.

Zum Warmwerden...

Der Reguläre Ausdruck $ab*c*a$ ...

1. ... beschreibt eine Typ-3 Sprache.
2. ... beschreibt die Sprache $L_1 = \{ab^n c^n a | n \in \mathbb{N}_0\}$
3. ... beschreibt eine endliche Sprache.

Rechtslineare Grammatiken ...

1. ... beschreiben reguläre Sprachen.
2. ... dürfen ϵ-Produktionen enthalten.
3. ... lassen sich in einen endlichen Automaten überführen.

Zum Warmwerden...

Der Reguläre Ausdruck $ab*c*a$ ...

1. ... beschreibt eine Typ-3 Sprache.
2. ... beschreibt die Sprache $L_1 = \{ab^n c^n a | n \in \mathbb{N}_0\}$
3. ... beschreibt eine endliche Sprache.

Rechtslineare Grammatiken ...

1. ... beschreiben reguläre Sprachen.
2. ... dürfen ϵ-Produktionen enthalten.
3. ... lassen sich in einen endlichen Automaten überführen.

Zum Warmwerden...

Für die Typen-Hierachie von Grammatiken gilt, ...

1. ... alle Grammatiken lassen sich als regulärer Ausdruck angeben
2. ... reguläre Sprachen sind die umfangreichsten.
3. ... wenn die Grammatik G Typ-2 ist (also eine KFG), so ist $L(G)$ auch Typ-2.

Ein Akzeptor, bei dem der Startzustand auch akzeptierender Zustand ist ...

1. ... akzeptiert alle Worte $w \in A$.
2. ... akzeptiert immer auch das leere Wort ϵ.
3. ... akzeptiert alle Worte $w \in A^*$.

Zum Warmwerden...

Für die Typen-Hierachie von Grammatiken gilt, ...

1. ... alle Grammatiken lassen sich als regulärer Ausdruck angeben
2. ... reguläre Sprachen sind die umfangreichsten.
3. ... wenn die Grammatik G Typ-2 ist (also eine KFG), so ist $L(G)$ auch Typ-2.

Ein Akzeptor, bei dem der Startzustand auch akzeptierender Zustand ist ...

1. ... akzeptiert alle Worte $w \in A$.
2. ... akzeptiert immer auch das leere Wort ϵ.
3. ... akzeptiert alle Worte $w \in A^*$.

Zum Warmwerden...

Für die Typen-Hierachie von Grammatiken gilt, ...

1. ... alle Grammatiken lassen sich als regulärer Ausdruck angeben
2. ... reguläre Sprachen sind die umfangreichsten.
3. ... wenn die Grammatik G Typ-2 ist (also eine KFG), so ist $L(G)$ auch Typ-2.

Ein Akzeptor, bei dem der Startzustand auch akzeptierender Zustand ist ...

1. ... akzeptiert alle Worte $w \in A$.
2. ... akzeptiert immer auch das leere Wort ϵ.
3. ... akzeptiert alle Worte $w \in A^*$.

Übersicht

Aufgabenblatt 11

Blatt 11

- Abgaben: 18 / 18
- Punkte: 11,1/20

Probleme

- bei jedem Automaten Startzustand angeben

Aufgabenblatt 11

Blatt 11

- Abgaben: 18 / 18
- Punkte: 11,1/20

Probleme

- bei jedem Automaten Startzustand angeben
- zu jedem Akzeptor die akzeptierenden Zustände angeben

Übersicht

Aufgabenblatt 12

Blatt 12

- Abgabe: 25.01.2013 um 12:30 Uhr im Untergeschoss des Infobaus
- Punkte: maximal 20

Themen

- Strukturelle Induktion

Aufgabenblatt 12

Blatt 12

- Abgabe: 25.01.2013 um 12:30 Uhr im Untergeschoss des Infobaus
- Punkte: maximal 20

Themen

- Strukturelle Induktion
- Akzeptoren, reguläre Ausdrücke und rechtslineare Grammatiken

Aufgabenblatt 12

Blatt 12

- Abgabe: 25.01.2013 um 12:30 Uhr im Untergeschoss des Infobaus
- Punkte: maximal 20

Themen

- Strukturelle Induktion
- Akzeptoren, reguläre Ausdrücke und rechtslineare Grammatiken
- Turing Maschinen

Aufgabenblatt 12

Blatt 12

- Abgabe: 25.01.2013 um 12:30 Uhr im Untergeschoss des Infobaus
- Punkte: maximal 20

Themen

- Strukturelle Induktion
- Akzeptoren, reguläre Ausdrücke und rechtslineare Grammatiken
- Turing Maschinen
- Turing Maschinen bauen

Aufgabenblatt 12

Blatt 12

- Abgabe: 25.01.2013 um 12:30 Uhr im Untergeschoss des Infobaus
- Punkte: maximal 20

Themen

- Strukturelle Induktion
- Akzeptoren, reguläre Ausdrücke und rechtslineare Grammatiken
- Turing Maschinen
- Turing Maschinen bauen
- Turing Maschinen verstehen

Übersicht

Definition

Reguläre Ausdrücke sind eine verbreitete und geeignete Notation, um reguläre Sprachen zu formalisieren.

Die Regeln (= Metazeichen)

Metazeichen	**Bedeutung**
$()$	Klammerung von Alternativen
$*$	n-maliges Vorkommen
$\|$	trennt Alternativen

Es gelten folgende Vorrangregeln:

- $*$ bindet stärker als Verkettung
- Verkettung (RS) bindet stärker als „oder“ $(R|S)$
- Überflüssige Klammern dürfen wir weglassen.
  So sind $(RS), ((RS)), \ldots$ und RS äquivalent

Definition

Die Sprache von R

Wenn **R** ein regulärer Ausdruck ist, dann bezeichnen wir mit $\langle R \rangle$ die Sprache, die dieser erzeugt.

- $\langle \emptyset \rangle = \{\}$
- Für $a \in A$ ist $\langle a \rangle = \{a\}$
- $\langle R_1 | R_2 \rangle = \langle R_1 \rangle \cup \langle R_2 \rangle$
- $\langle R_1 R_2 \rangle = \langle R_1 \rangle \cdot \langle R_2 \rangle$

R_1 und R_2 sind hier zwei beliebige reguläre Ausdrücke.

Die Sprache von R

Beispiel 1

Welche Wörter erzeugt der folgende reguläre Ausdruck R?

- $R = (a|b) * abb(a|b)*$?

Die Sprache von R

Beispiel 1

Welche Wörter erzeugt der folgende reguläre Ausdruck R?

- $R = (a|b) * abb(a|b)*$?
- $\langle R \rangle$ enthält genau die Wörter, in denen das Teilwort *abb* vorkommt.

Beispiel 2

Gebe einen regulären Ausdruck für die Sprache aller Wörter die nicht *ab* enthalten

Die Sprache von R

Beispiel 1

Welche Wörter erzeugt der folgende reguläre Ausdruck R?

- $R = (a|b) * abb(a|b)*$?
- $\langle R \rangle$ enthält genau die Wörter, in denen das Teilwort abb vorkommt.

Beispiel 2

Gebe einen regulären Ausdruck für die Sprache aller Wörter die nicht ab enthalten

- $b*a*$

Die Sprache von R

Karlsruhe Institute of Technology

Beispiel 3

Welcher reguläre Ausdruck R erzeugt die Sprache $\{\epsilon\}$?

Die Sprache von R

Beispiel 3

Welcher reguläre Ausdruck R erzeugt die Sprache $\{\epsilon\}$?

- $\emptyset*$, denn $\langle\emptyset\rangle^* = \{\}^* = \{\epsilon\}$

Die Sprache von R

Beispiel 3

Welcher reguläre Ausdruck R erzeugt die Sprache $\{\epsilon\}$?

- $\emptyset*$, denn $\langle\emptyset\rangle^* = \{\}^* = \{\epsilon\}$

Beispiel 4

Gebe einen regulären Ausdruck für die Sprache aller Wörter mit mindestens 3 b's an!

Die Sprache von R

Beispiel 3

Welcher reguläre Ausdruck R erzeugt die Sprache $\{\epsilon\}$?

- $\emptyset*$, denn $\langle\emptyset\rangle^* = \{\}^* = \{\epsilon\}$

Beispiel 4

Gebe einen regulären Ausdruck für die Sprache aller Wörter mit mindestens 3 b's an!

- $(a|b) * b(a|b) * b(a|b) * b(a|b)*$ oder

Die Sprache von R

Beispiel 3

Welcher reguläre Ausdruck R erzeugt die Sprache $\{\epsilon\}$?

- $\emptyset*$, denn $\langle\emptyset\rangle^* = \{\}^* = \{\epsilon\}$

Beispiel 4

Gebe einen regulären Ausdruck für die Sprache aller Wörter mit mindestens 3 b's an!

- $(a|b) * b(a|b) * b(a|b) * b(a|b)*$ oder
- $a * ba * ba * b(a|b)*$

Übersicht

rechtslineare Grammatiken

Karlsruhe Institute of Technology

etwas genauer...

Eine rechtslineare Grammatik ist eine kontextfreie Grammatik $G = (N, T, S, P)$ mit folgenden Einschränkungen. Jede Produktion ist entweder von der Form

- $X \to w$ oder
- $X \to wY$ mit $w \in T^*$ und $X, Y \in N$

rechtslineare Grammatiken

Karlsruhe Institute of Technology

etwas genauer...

Eine rechtslineare Grammatik ist eine kontextfreie Grammatik $G = (N, T, S, P)$ mit folgenden Einschränkungen. Jede Produktion ist entweder von der Form

- $X \to w$ oder
- $X \to wY$ mit $w \in T^*$ und $X, Y \in N$

Regex

Zu jeder rechtslinearen Grammatik gibt es:

- ...einen entsprechenden regulären Ausdruck und

rechtslineare Grammatiken

Karlsruhe Institute of Technology

etwas genauer...

Eine rechtslineare Grammatik ist eine kontextfreie Grammatik $G = (N, T, S, P)$ mit folgenden Einschränkungen. Jede Produktion ist entweder von der Form

- $X \to w$ oder
- $X \to wY$ mit $w \in T^*$ und $X, Y \in N$

Regex

Zu jeder rechtslinearen Grammatik gibt es:

- ... einen entsprechenden regulären Ausdruck und
- ... einen einen deterministischen endlichen Automaten

Zu jeder rechtslinearen gibt es äquivalente linkslineare Grammatiken. Diese „können“ nichts anderes als rechtslineare Grammatiken, daher ignorieren wir sie in dieser Vorlesung.

rechtslineare Grammatiken

Ein Beispiel

Gegeben Sei die Grammatik
$G = (\{X, Y\}, \{a, b\}, X, \{X \rightarrow aY|\epsilon, Y \rightarrow Xb\})$

- Ist diese Grammatik rechtslinear?

rechtslineare Grammatiken

Ein Beispiel - oder auch nicht...

Gegeben Sei die Grammatik
$G = (\{X, Y\}, \{a, b\}, X, \{X \to aY|\epsilon, Y \to Xb\})$

- Ist diese Grammatik rechtslinear?
  G ist offensichtlich nicht rechtslinear, denn die Produktion $Y \to Xb$ hat das Nichtterminalsymbol links vom Terminalsymbol
  (Die Produktion ist linkslinear)!

rechtslineare Grammatiken

Ein Beispiel - oder auch nicht...

Gegeben Sei die Grammatik
$G = (\{X, Y\}, \{a, b\}, X, \{X \to aY | \epsilon, Y \to Xb\})$

- Ist diese Grammatik rechtslinear?
  G ist offensichtlich nicht rechtslinear, denn die Produktion $Y \to Xb$ hat das Nichtterminalsymbol links vom Terminalsymbol
  (Die Produktion ist linkslinear)!
- Die Grammatik erzeugt die Sprache $L(G) = \{a^k b^k | k \in \mathbb{N}_0\}$

rechtslineare Grammatiken

Ein Beispiel - oder auch nicht...

Gegeben Sei die Grammatik
$G = (\{X, Y\}, \{a, b\}, X, \{X \to aY | \epsilon, Y \to Xb\})$

- Ist diese Grammatik rechtslinear?
  G ist offensichtlich nicht rechtslinear, denn die Produktion $Y \to Xb$ hat das Nichtterminalsymbol links vom Terminalsymbol
  (Die Produktion ist linkslinear)!
- Die Grammatik erzeugt die Sprache $L(G) = \{a^k b^k | k \in \mathbb{N}_0\}$
- Kann es eine rechtslineare Grammatik für diese Sprache geben?
  Ist diese Sprache regulär?

rechtslineare Grammatiken

Ein Beispiel - oder auch nicht...

Gegeben Sei die Grammatik
$G = (\{X, Y\}, \{a, b\}, X, \{X \to aY | \epsilon, Y \to Xb\})$

- Ist diese Grammatik rechtslinear?
  G ist offensichtlich nicht rechtslinear, denn die Produktion $Y \to Xb$ hat das Nichtterminalsymbol links vom Terminalsymbol
  (Die Produktion ist linkslinear)!
- Die Grammatik erzeugt die Sprache $L(G) = \{a^k b^k | k \in \mathbb{N}_0\}$
- Kann es eine rechtslineare Grammatik für diese Sprache geben?
  Ist diese Sprache regulär?
  Nein, ist sie nicht!

von G zu L(G)

Aufgabe

Betrachte $G = (\{X, Y, Z\}, \{a, b\}, X, P)$ mit
$P = \{X \to aX|bY|\epsilon, Y \to aX|bZ|\epsilon, Z \to aZ|bZ\}$

- Was ist $L(G)$?
- Geben Sie einen endlichen Automaten an, der L(G) akzeptiert.
- Lässt sich diese Grammatik noch vereinfachen?

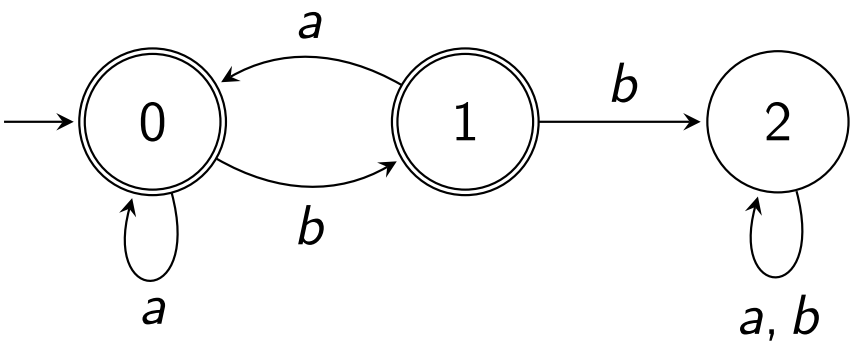
von G zu L(G)

Aufgabe

Betrachte $G = (\{X, Y, Z\}, \{a, b\}, X, P)$ mit
$P = \{X \to aX|bY|\epsilon, Y \to aX|bZ|\epsilon, Z \to aZ|bZ\}$

- Was ist $L(G)$?
- Geben Sie einen endlichen Automaten an, der L(G) akzeptiert.
- Lässt sich diese Grammatik noch vereinfachen?

Lösung



Ist doch alles das Gleiche, oder?

Gleiche Sprache - andere Grammatik

Folgende Grammatiken erzeugen die gleiche Sprache

- $G = (\{X, Y, Z\}, \{a, b\}, X, P)$ mit
  $P = \{X \to aX|bY|\epsilon, Y \to aX|bZ|\epsilon, Z \to aZ|bZ\}$

Ist doch alles das Gleiche, oder?

Gleiche Sprache - andere Grammatik

Folgende Grammatiken erzeugen die gleiche Sprache

- $G = (\{X, Y, Z\}, \{a, b\}, X, P)$ mit
  $P = \{X \to aX|bY|\epsilon, Y \to aX|bZ|\epsilon, Z \to aZ|bZ\}$
- $G = (\{X, Y\}, \{a, b\}, X, P)$ mit $P = \{X \to aX|bY|\epsilon, Y \to aX|\epsilon\}$

Ist doch alles das Gleiche, oder?

Gleiche Sprache - andere Grammatik

Folgende Grammatiken erzeugen die gleiche Sprache

- $G = (\{X, Y, Z\}, \{a, b\}, X, P)$ mit
  $P = \{X \to aX|bY|\epsilon, Y \to aX|bZ|\epsilon, Z \to aZ|bZ\}$
- $G = (\{X, Y\}, \{a, b\}, X, P)$ mit $P = \{X \to aX|bY|\epsilon, Y \to aX|\epsilon\}$
- $G = (\{X\}, \{a, b\}, X, P)$ mit $P = \{X \to aX|baX|b|\epsilon\}$

Übersicht

Karlsruhe Institute of Technology

Kantorowitsch-Bäume, Regex-Bäume

Kantorowitsch

Beispiel:

- wie sieht der Baum zu $3 + (a + b) * (-c)$ aus?

Kantorowitsch-Bäume, Regex-Bäume

Kantorowitsch

Beispiel:

- wie sieht der Baum zu $3 + (a + b) * (-c)$ aus?
- wie hoch ist der Baum?

Kantorowitsch-Bäume, Regex-Bäume

Kantorowitsch

Beispiel:

- wie sieht der Baum zu $3 + (a + b) * (-c)$ aus?
- wie hoch ist der Baum? Antwort: 3

Kantorowitsch-Bäume, Regex-Bäume

Kantorowitsch

Beispiel:

- wie sieht der Baum zu $3 + (a + b) * (-c)$ aus?
- wie hoch ist der Baum? Antwort: 3

Die Höhe eines Baumes entspricht auch dem längsten wiederholungsfreien Weg von der Wurzel zu den Blättern.

Kantorowitsch-Bäume zu Reg. Ausdrücken

Kantorowitsch-Bäume, Regex-Bäume

Kantorowitsch

Beispiel:

- wie sieht der Baum zu $3 + (a + b) * (-c)$ aus?
- wie hoch ist der Baum? Antwort: 3

Die Höhe eines Baumes entspricht auch dem längsten wiederholungsfreien Weg von der Wurzel zu den Blättern.

Kantorowitsch-Bäume zu Reg. Ausdrücken

- Wird ein regulärer Ausdruck als Baum dargestellt, sind Klammern auch unnötig

Kantorowitsch-Bäume, Regex-Bäume

Kantorowitsch

Beispiel:

- wie sieht der Baum zu $3 + (a + b) * (-c)$ aus?
- wie hoch ist der Baum? Antwort: 3

Die Höhe eines Baumes entspricht auch dem längsten wiederholungsfreien Weg von der Wurzel zu den Blättern.

Kantorowitsch-Bäume zu Reg. Ausdrücken

- Wird ein regulärer Ausdruck als Baum dargestellt, sind Klammern auch unnötig
- durch die „Eltern-Kind“- bzw. „ist-Teilausdruck“-Beziehung im Baum ist die Klammerung eindeutig festgelegt

Übersicht

Typische Fragestellung

- Gibt es Aufgaben, die von einem Rechner unabhängig
  - von der Art der Programmierung
  - von physikalischen und elektronischen Beschränkungen
- nicht gelöst werden können?
- **Hauptfrage:** Welche Probleme sind dann berechenbar?

Typische Fragestellung

- Gibt es Aufgaben, die von einem Rechner unabhängig
  - von der Art der Programmierung
  - von physikalischen und elektronischen Beschränkungen
- nicht gelöst werden können?
- **Hauptfrage:** Welche Probleme sind dann berechenbar?

Lösungsansätze

Entwicklung neuer Konzepte wie

- eine grundlegende **Problemformulierung**
- ein grundlegendes **Rechnermodell**

Dafür muss geklärt werden wie ein Rechner (der nur *Nullen* und *Einsen* kennt) ein Problem überhaupt löst.

Suche nach Rechnermodell

Suche nach Rechnermodell

Karlsruhe Institute of Technology

Vorschlag

endliche Automaten als Grundlage für „allgemeine“ theoretische Aussagen über „Berechenbarkeit“

Suche nach Rechnermodell

Vorschlag

endliche Automaten als Grundlage für „allgemeine“ theoretische Aussagen über „Berechenbarkeit“

Problem

- nicht mächtig genug!
- es ist zwar erfassbar, ob ein Getränkeautomat (wie aus der Vorlesung) eine Eingabe verabeiten kann, ABER das Modell wäre spätestens bei kontextsensitiven Sprachen überfordert

Suche nach Rechnermodell

Karlsruhe Institute of Technology

Vorschlag

endliche Automaten als Grundlage für „allgemeine“ theoretische Aussagen über „Berechenbarkeit“

Problem

- nicht mächtig genug!
- es ist zwar erfassbar, ob ein Getränkeautomat (wie aus der Vorlesung) eine Eingabe verabeiten kann, ABER das Modell wäre spätestens bei kontextsensitiven Sprachen überfordert

Frage: Gibt es ein mächtigeres, realistisches Rechnermodell, das geeignet ist?

Übersicht

Aufbau der Turing-Maschine

Erfinder: Alan Turing (1936)

Aufbau der Turing-Maschine

Erfinder: Alan Turing (1936)

Bestandteile

- beidseitig unendliches Eingabe- und Rechenband
- freibeweglicher Lese-/Schreibkopf
- gesteuert von einer endlichen Kontrolle

Aufbau der Turing-Maschine

Erfinder: Alan Turing (1936)

Bestandteile

- beidseitig unendliches Eingabe- und Rechenband
- freibeweglicher Lese-/Schreibkopf
- gesteuert von einer endlichen Kontrolle

Eigenschaften

- Eingabe- und Rechenband enthält eine Folge von Symbolen
- Kontrolle ist in einem von endlich vielen Zuständen
- Zellen des Bandes enthalten jeweils höchstens ein Symbol aus dem Bandalphabet

Funktionsweise

So arbeitet die TM

Ist die Turing-Maschine (TM) in einem bestimmten Zustand und liest ein Symbol...

- so geht sie in einen Folgezustand über
- überschreibt eventuell das Symbol und
- bewegt den Lese-/Schreibkopf eine Stelle nach rechts, nach links oder überhaupt nicht

Definition der TM

Ganz genau

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

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

Definition der TM

Ganz genau

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

- eine endliche **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 $m : Z \times X \dashrightarrow \{-1, 0, 1\}$

Definition der TM

Anmerkungen

- Die Funktionen **f**, **g** und **m** beschreiben, wie das eingelesene Zeichen verarbeitet werden soll (gemeinsamer Definitionsbereich)

Definition der TM

Anmerkungen

- Die Funktionen **f**, **g** und **m** beschreiben, wie das eingelesene Zeichen verarbeitet werden soll (gemeinsamer 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

Definition der TM

Anmerkungen

- Die Funktionen **f**, **g** und **m** beschreiben, wie das eingelesene Zeichen verarbeitet werden soll (gemeinsamer 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
- Im Bandalphabet gibt es ein sogenanntes Blanksymbol $\square \in X$ zur Beschreibung einer leeren Zelle auf dem Band
- ein endliches Eingabealphabet $A \subset X \setminus \{\square\}$
- eine endliche Menge von akzpetierenden Zuständen $F \subseteq Z$

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

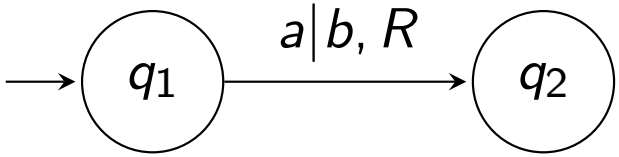
Eine Turingmaschine befindet sich zu jedem Zeitpunkt in einem *Gesamtzustand*, der als **Konfiguration** $(z, b, p) \in Z \times X^{\mathbb{Z}} \times \mathbb{Z}$ bezeichnet wird

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**.

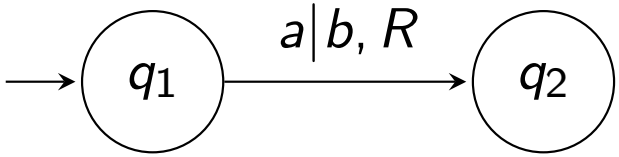
Beispiele

Einfach I



Beispiele

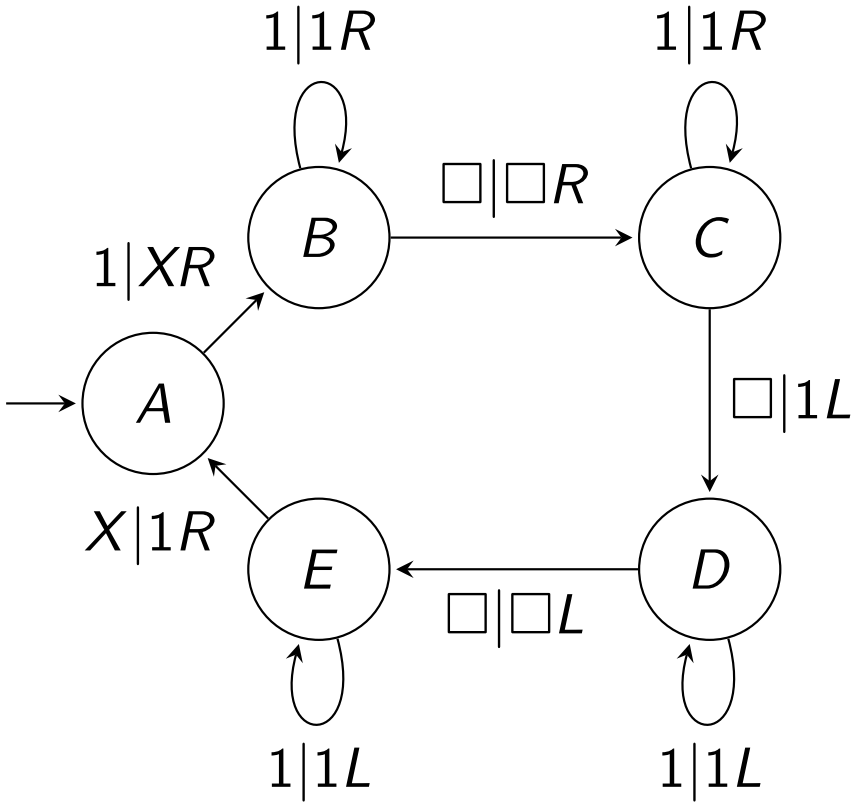
Einfach I



Bedeutung des Übergangs

Ist die TM im Zustand q_1 und liest das Sysmbol a, so überschreit sie dieses a mit b, geht auf dem Bande eine Stelle nach rechts und wechselt in den Zustand q_2

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

Verhalten von TM bei Entscheidungsproblemen

Karlsruhe Institute of Technology

akzeptierende Zustände

Eine Turing-Maschine akzeptiert eine Eingabe w, wenn sie nach Lesen von w in einem akzeptierenden Zustand $F \subset Z$ stoppt.
Es gibt Eingaben, für die eine Turing-Maschine unter Umständen niemals stoppt, bzw. nur in einem nicht akzeptierenden Zustand.

Verhalten von TM bei Entscheidungsproblemen

akzeptierende Zustände

Eine Turing-Maschine akzeptiert eine Eingabe w, wenn sie nach Lesen von w in einem akzeptierenden Zustand $F \subset Z$ stoppt.
Es gibt Eingaben, für die eine Turing-Maschine unter Umständen niemals stoppt, bzw. nur in einem nicht akzeptierenden Zustand.

Terminierung

Wann terminiert eine TM?

Verhalten von TM bei Entscheidungsproblemen

akzeptierende Zustände

Eine Turing-Maschine akzeptiert eine Eingabe w, wenn sie nach Lesen von w in einem akzeptierenden Zustand $F \subset Z$ stoppt.
Es gibt Eingaben, für die eine Turing-Maschine unter Umständen niemals stoppt, bzw. nur in einem nicht akzeptierenden Zustand.

Terminierung

Wann terminiert eine TM?
Wenn der nächste Schritt nicht mehr definiert ist.

Ihr seid dran...

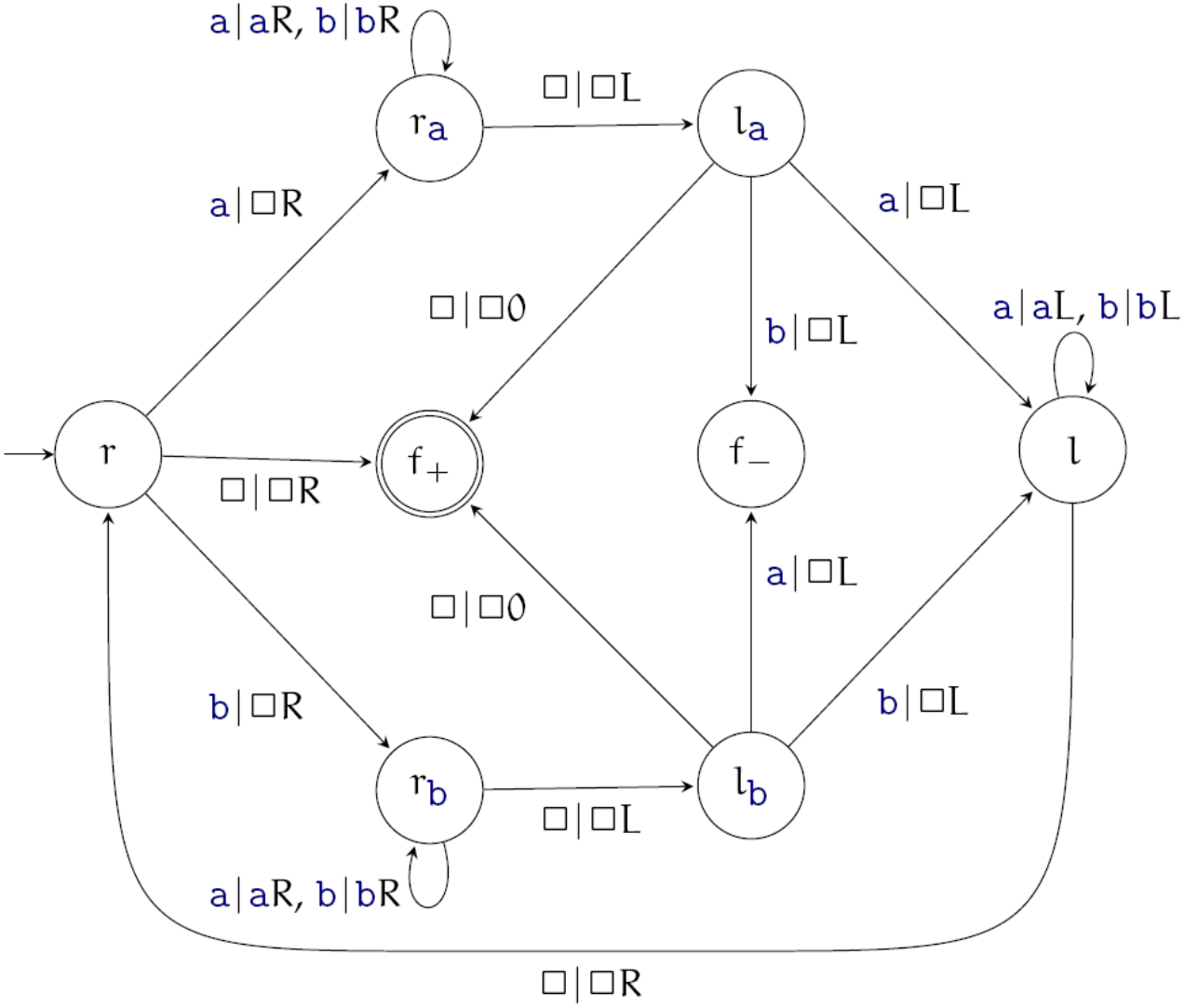
Karlsruhe Institute of Technology

Aufgabe

Gebt die Berechnung der folgenden Turingmaschine für die Eingaben

1. aba
2. baba

Ihr seid dran...



Ihr seid dran...

Karlsruhe Institute of Technology

Aufgabe

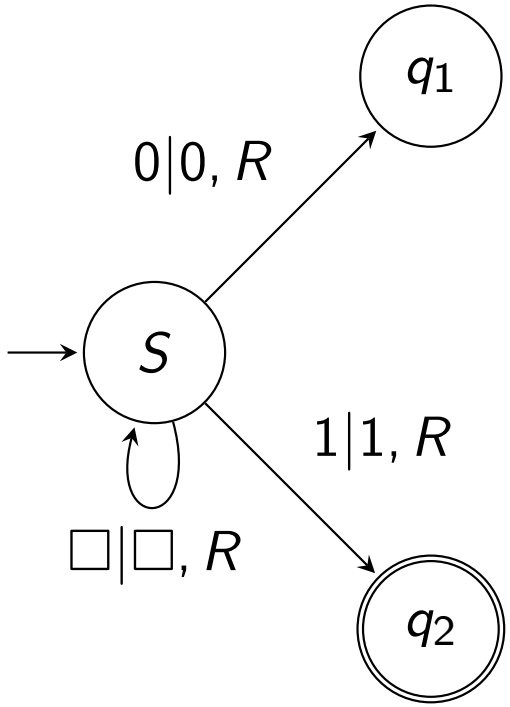
Gebt eine Turingmaschine an die alle Wörter aus $\{0, 1\}^*$ erkennt, die mit einer Eins beginnen

Ihr seid dran...

Aufgabe

Gebt eine Turingmaschine an die alle Wörter aus $\{0, 1\}^*$ erkennt, die mit einer Eins beginnen

Lösung (nicht minimal)



partielle Funktion von TM

Karlsruhe Institute of Technology

Eine TM kann mehr...

Eine Turing-Maschine erkennt nicht nur Mengen von Wörtern (Sprachen), sondern sie verändert auch die Eingabe und hat insofern auch eine Ausgabe (= Inhalt des Bandes nach der Bearbeitung).
Eine TM kann so zur Berechnung von Funktionen genutzt werden, wie der Addition oder dem Vergleich zweier Binärzahlen

Wir konstruieren eine solche TM...

Ihr seid dran...

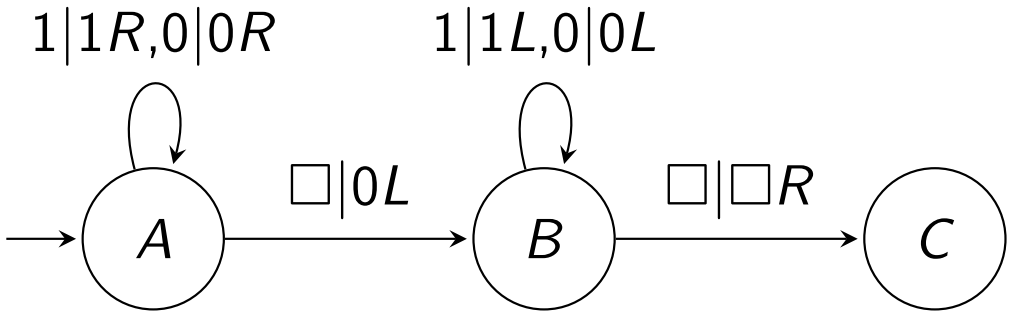
Gebt eine Turing-Maschine an, die eine in Binärcodierung gegebene natürliche Zahl $n \geq 1$ mit der Zahl 2 multipliziert. Der Schreib-/Lesekopf soll sich zu Anfang und Ende der Berechnung jeweils auf dem ersten Bit (der größten Zweierpotenz zugeordnet) der Darstellung von n befinden...

Wir konstruieren eine solche TM...

Ihr seid dran...

Gebt eine Turing-Maschine an, die eine in Binärcodierung gegebene natürliche Zahl $n \geq 1$ mit der Zahl 2 multipliziert. Der Schreib-/Lesekopf soll sich zu Anfang und Ende der Berechnung jeweils auf dem ersten Bit (der größten Zweierpotenz zugeordnet) der Darstellung von n befinden...

Lösung



Die „verschiedenen Arten“ von TMs

Verknüpfung von Entscheidbarkeit von Sprachen und der Berechenbarkeit von Funktionen:

- Eine Turing-Maschine akzeptiert eine Sprache L, wenn sie genau für die Eingaben $w \in L$ stoppt.
- L ist entscheidbar, wenn es eine Turing-Maschine gibt, die auf allen Eingaben stoppt und L akzeptiert.
- Die Funktion f heißt berechenbar, wenn eine Turing-Maschine existiert, die f realisiert.

Berechnungsschritt, d.h. Konfigurationswechsel

Berechnungsschritt, d.h. Konfigurationswechsel

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 $0 < i$ gilt $c_i = \Delta_1(c_{i-1})$
- **nicht haltend**

Turingmaschinenakzeptoren

Turingmaschinenakzeptoren

Karlsruhe Institute of Technology

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

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

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

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...

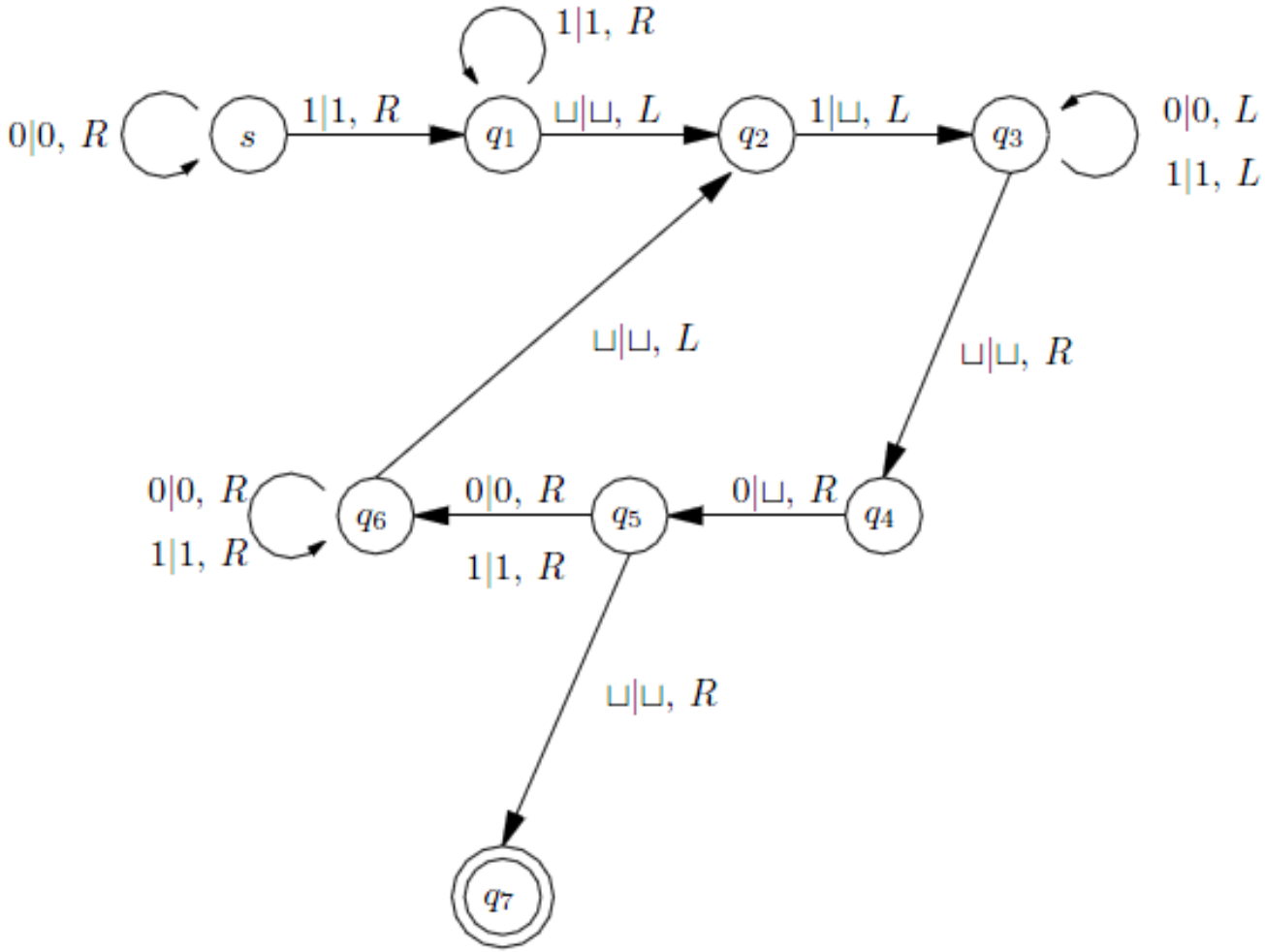
Karlsruhe Institute of Technology

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

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

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

Zum Schluss...

Was ihr nun wissen solltet!

Zum Schluss...

Karlsruhe Institute of Technology

Was ihr nun wissen solltet!

- Was verstehen wir unter Berechenbarkeit?

Zum Schluss...

Karlsruhe Institute of Technology

Was ihr nun wissen solltet!

- Was verstehen wir unter Berechenbarkeit?
- Welches Rechenmodell ist in der Informatik von zentrale Bedeutung?

Zum Schluss...

Karlsruhe Institute of Technology

Was ihr nun wissen solltet!

- Was verstehen wir unter Berechenbarkeit?
- Welches Rechenmodell ist in der Informatik von zentrale Bedeutung?
- Was sind die Bestandteile dieses Rechenmodells? Und wie arbeitet es?

Zum Schluss...

Karlsruhe Institute of Technology

Was ihr nun wissen solltet!

- Was verstehen wir unter Berechenbarkeit?
- Welches Rechenmodell ist in der Informatik von zentrale Bedeutung?
- Was sind die Bestandteile dieses Rechenmodells? Und wie arbeitet es?
- Was lässt sich aus diesem Rechenmodell ableiten?

Zum Schluss...

Karlsruhe Institute of Technology

Was ihr nun wissen solltet!

- Was verstehen wir unter Berechenbarkeit?
- Welches Rechenmodell ist in der Informatik von zentrale Bedeutung?
- Was sind die Bestandteile dieses Rechenmodells? Und wie arbeitet es?
- Was lässt sich aus diesem Rechenmodell ableiten?
- Wann ist eine Sprache entscheidbar?

Zum Schluss...

Karlsruhe Institute of Technology

Was ihr nun wissen solltet!

- Was verstehen wir unter Berechenbarkeit?
- Welches Rechenmodell ist in der Informatik von zentrale Bedeutung?
- Was sind die Bestandteile dieses Rechenmodells? Und wie arbeitet es?
- Was lässt sich aus diesem Rechenmodell ableiten?
- Wann ist eine Sprache entscheidbar?

Ihr wisst was nicht?

Stellt **jetzt** Fragen!

Ende


PANTOMIMEN-LEMMINGE
DAS GIBT ES DOCH NICHT. HIER GEHT ES AUCH NICHT WEITER.
DIESE VERDAMMTEN UNSICHTBAREN WÄNDE.
WWW.NICHTLUSTIG.DE