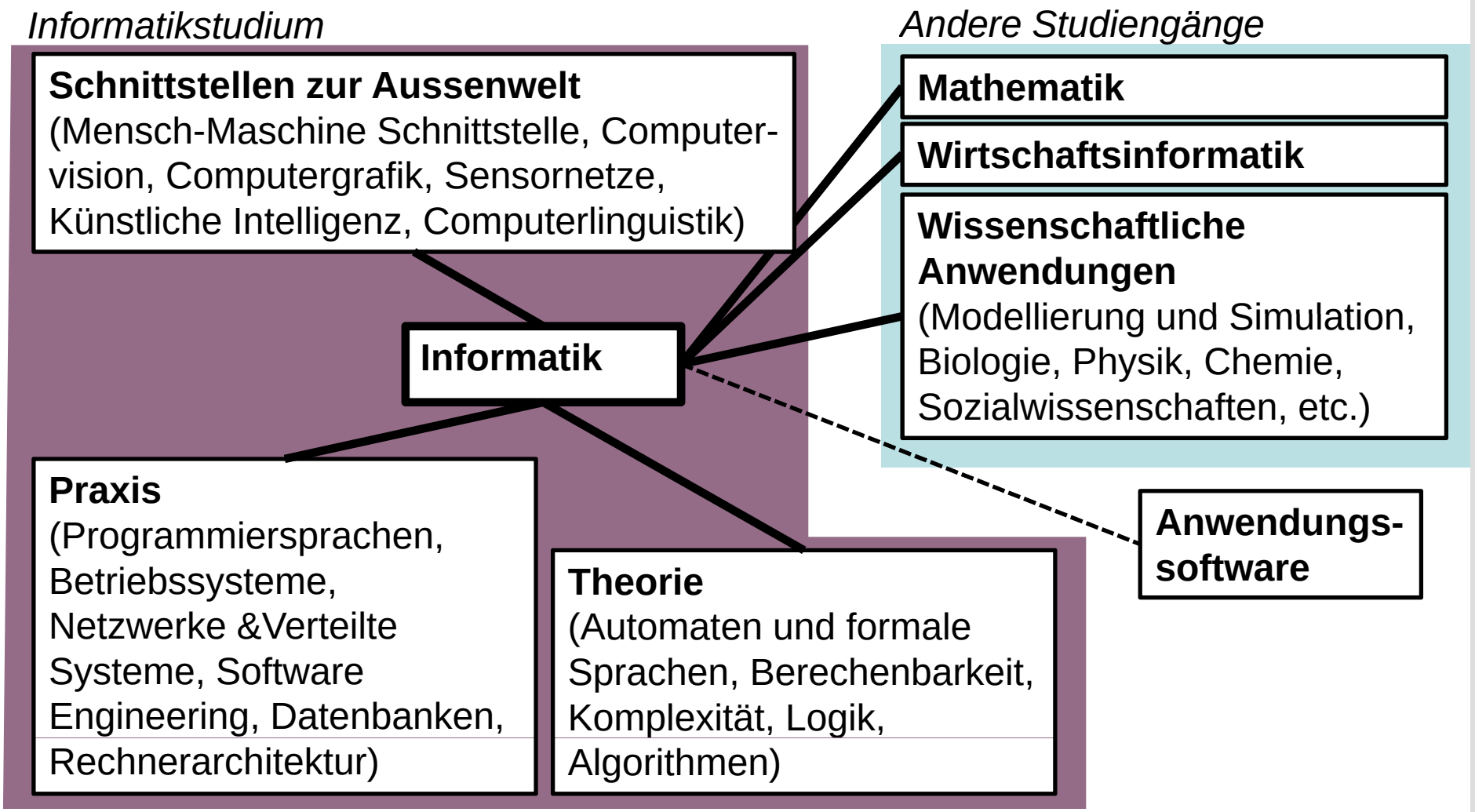
Einführung in die Informatik

Endliche Automaten, reguläre Sprachen

Dr. Thomas Studer

Übersicht


Informatikstudium
Andere Studiengänge
Schnittstellen zur Aussenwelt
(Mensch-Maschine Schnittstelle, Computer-vision, Computergrafik, Sensornetze, Künstliche Intelligenz, Computerlinguistik)
Mathematik
Wirtschaftsinformatik
Wissenschaftliche Anwendungen
(Modellierung und Simulation, Biologie, Physik, Chemie, Sozialwissenschaften, etc.)
Informatik
Anwendungs-software
Praxis
(Programmiersprachen, Betriebssysteme, Netzwerke &Verteilte Systeme, Software Engineering, Datenbanken, Rechnerarchitektur)
Theorie
(Automaten und formale Sprachen, Berechenbarkeit, Komplexität, Logik, Algorithmen)

Endliche Automaten

Ein **endlicher Automat** ist ein mathematisches Modell, bzw. eine **Abstraktion** eines „echten“ Automaten, der Informationen Zeichen für Zeichen einliest und das eingelesene Zeichen jeweils sofort verarbeitet.

Ein endlicher Automat besitzt eine endliche Menge von **Zuständen** und keinen zusätzlichen Speicher.

Ablauf eines Arbeitsschritts

Ausgangslage:

- Unter dem Lesekopf ist das Zeichen x
- Der Automat ist im Zustand q

Ablauf:

1. Der Lesekopf liest x
2. Der Schreibkopf schreibt das nächste Zeichen y auf das Ausgabeband
3. Der Zustand wechselt von q nach q‘ (evtl. q = q‘)
4. Das Eingabeband wird vorgerückt

Siehe Bild auf Tafel

Begriffe

Anfangszustand (Startzustand)

= Zustand vor dem Einlesen des 1. Zeichens
(es gibt nur einen Anfangszustand)

Endzustand

Es wird kein Zeichen mehr eingelesen

Die seit dem Start verarbeitete Folge von Eingabezeichen ist akzeptiert („korrekt“)

I. Allg. mehrere Endzustände

Es ist möglich, dass der Automat in einem Zustand anhält, der kein Endzustand ist. Die Folge von Eingabezeichen ist in diesem Fall **nicht** akzeptiert.

Mathematische Beschreibung

Ein endlicher Automat ist ein 6-Tupel

$$A = (I,O,Q,\delta,q_0,F)$$

mit

I: Input-Alphabet, aus endlich vielen Zeichen

O: Output-Alphabet, aus endlich vielen Zeichen

Q: Zustandsmenge, aus endlich vielen Zuständen

δ: Übergangsfunktion

$$\delta : Q \times (I \cup \{\varepsilon\}) \rightarrow Q \times (O \cup \{\varepsilon\})$$

q_0: Anfangszustand

F: Menge der Endzustände (F ist Teilmenge von Q)

Beispiel: Fussgängerampel

I = {T,TN},
mit T: Taste gedrückt, TN: Taste nicht gedrückt

O = {SG.SR}.
mit SG: Strasse grün, SR: Strasse rot

Q = {q,r}

q_0 = q

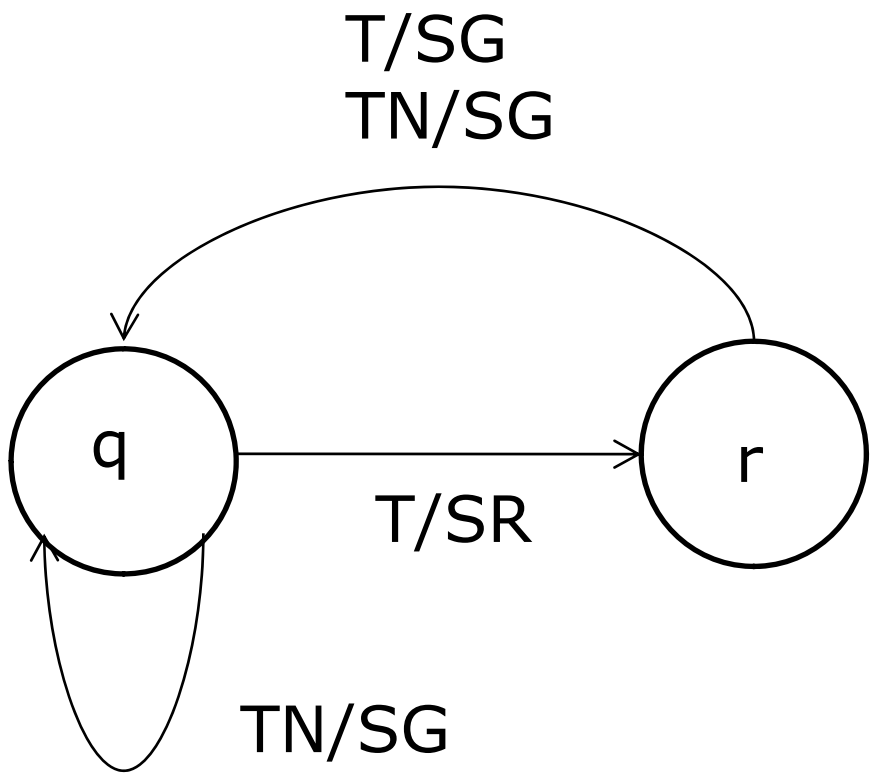
F = leere Menge

δ(q,T) := (r,SR)

δ(q,TN) := (q,SG)

δ(r,T) := (q,SG)

δ(r,TN) := (q,SG)



Beispiel: Getränkeautomat

Getränk kostet 2.50. Automat gibt Wechselgeld

I = {100,50}, wobei 100: 1 Franken, 50: 50 Rappen

O= {G,W}, wobei G: Getränk, W: Wechselgeld

Q={0,50,100,150,200,300}, merkt sich eingeworfenen Betrag

δ(0,50)	:= (50,ε)	δ(150,50)	:= (200,ε)
δ(0,100)	:= (100,ε)	δ(150,100)	:= (0,G)
δ(50,50)	:= (100,ε)	δ(200,50)	:= (0,G)
δ(50,100)	:= (150,ε)	δ(200,100)	:= (300,G)
δ(100,50)	:= (150,ε)	δ(300, ε)	:= (0,W)
δ(100,100)	:= (200,ε)		

Begriffe

Ein Automat heisst **nicht-deterministisch**, falls für einen Zustand q und ein Inputzeichen x mehrere Folgezustände q‘ möglich sind.

In diesem Fall gilt:

$$\delta: (q,x) \rightarrow \{(q_1,y_1),\ldots,(q_n,y_n)\}$$

Falls nach Abarbeitung des endlichen Eingabebandes der Automat in einem Endzustand ist, hat er das Eingabewort **akzeptiert**. Auf den Ausgabeband steht dann das zugehörige **Ausgabewort**.

Begriffe

Ein Automat heisst **übersetzend** (Transducer), falls er zu einem Eingabewort ein Ausgabewort berechnet.

Ein Automat heisst **erkennend**, falls er kein Ausgabeband verwendet, sondern einfach Eingabewörter akzeptiert oder nicht akzeptiert.

Beispiel: Transducer

Transducer soll Oktalzahlen in Binärzahlen übersetzen

I = {0,1,2,3,4,5,6,7} und O = {0,1}

Q={s,a,b,c,d,e,f}

$\delta(s,0) := (a,0)$ $\quad$ $\delta(a,\varepsilon) := (e,0)$

$\delta(s,1) := (b,0)$ $\quad$ $\delta(b,\varepsilon) := (f,0)$

$\delta(s,2) := (c,0)$ $\quad$ $\delta(c,\varepsilon) := (e,1)$

$\delta(s,3) := (d,0)$ $\quad$ $\delta(d,\varepsilon) := (f,1)$

$\delta(s,4) := (a,1)$ $\quad$ $\delta(e,\varepsilon) := (s,0)$

$\delta(s,5) := (b,1)$ $\quad$ $\delta(f,\varepsilon) := (s,1)$

$\delta(s,6) := (c,1)$

$\delta(s,7) := (d,1)$

Beispiel: Transducer

Transducer soll 2 einstellige Binärzahlen addieren

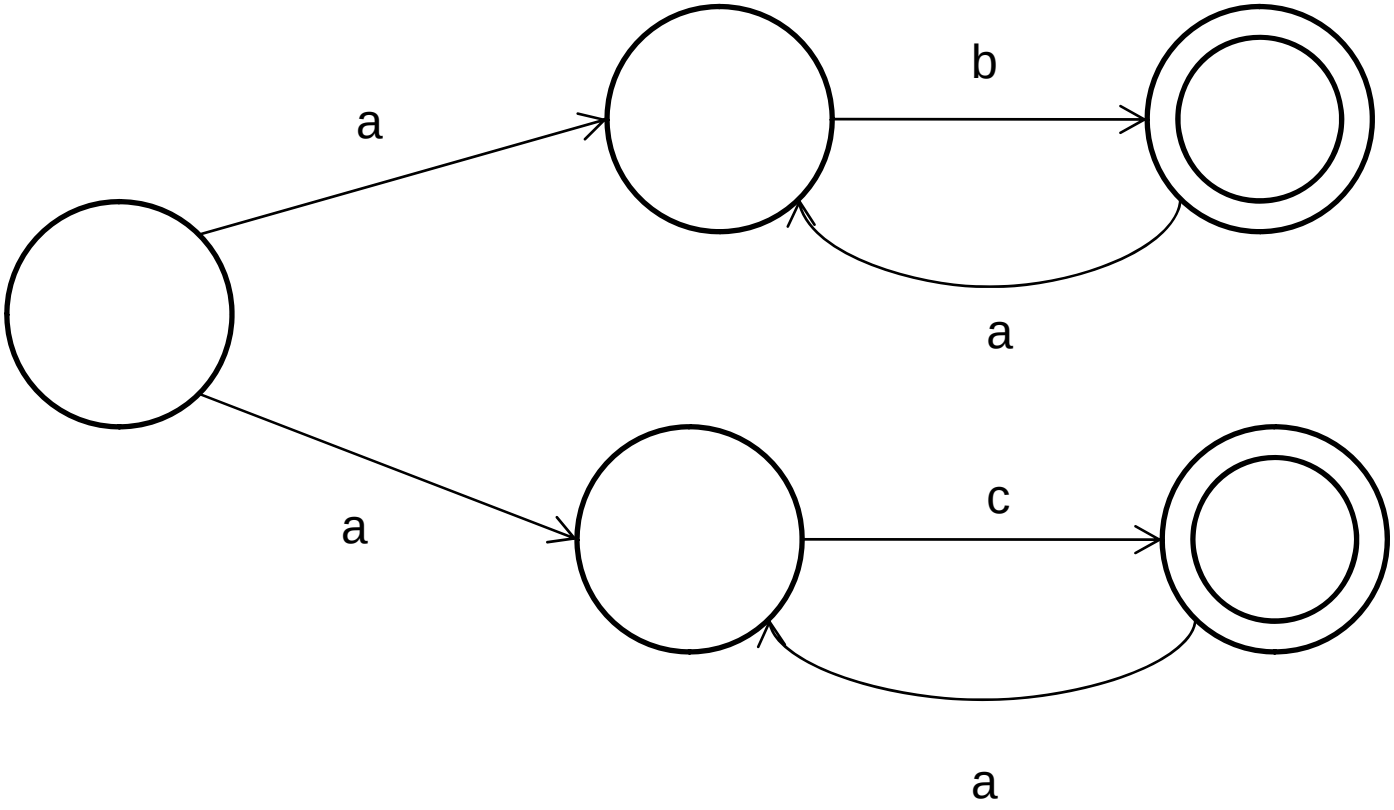
I = O = {0,1}

Q={s,0,1,2,3},

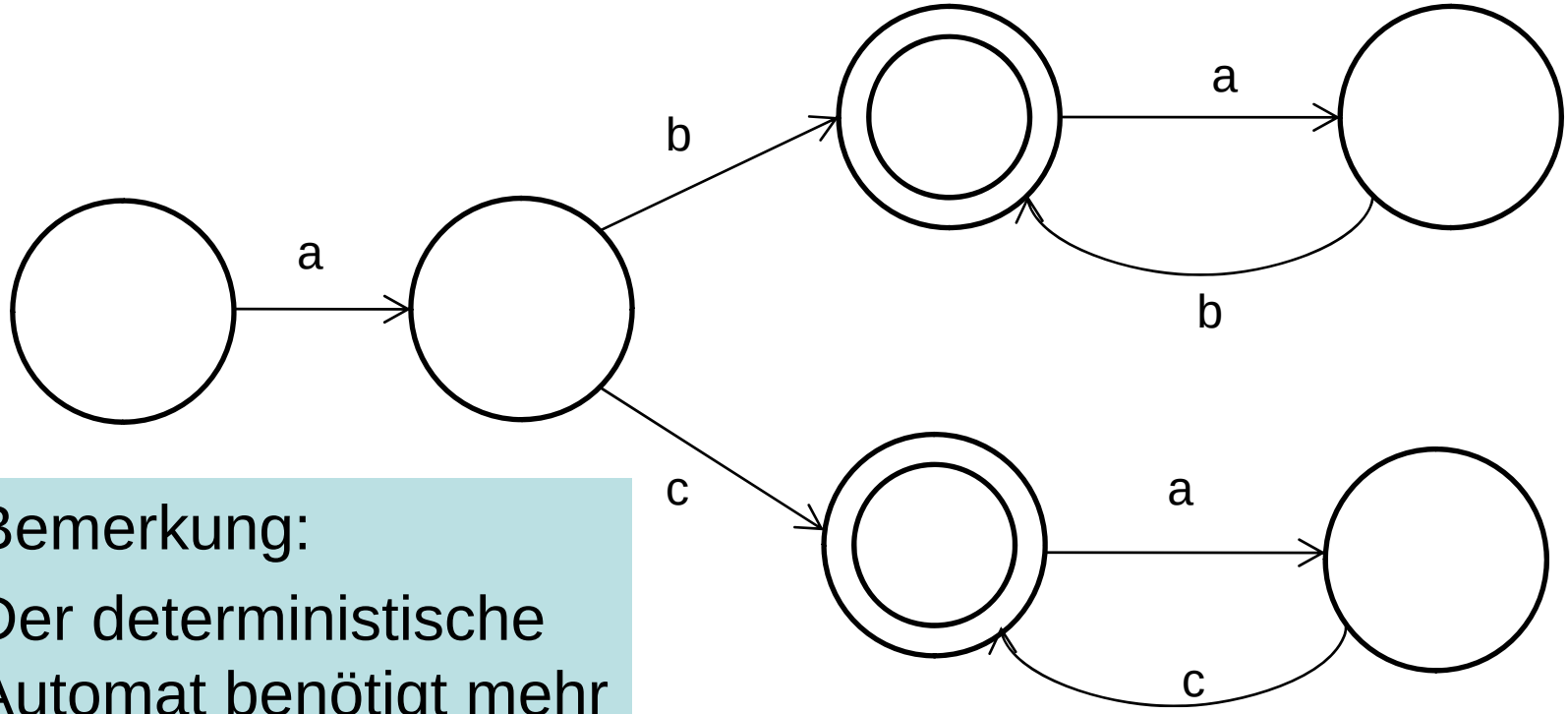
$\delta(s,0) := (0,\varepsilon)$	$\delta(1,0) := (3,0)$
$\delta(s,1) := (1,\varepsilon)$	$\delta(1,1) := (3,1)$
$\delta(0,0) := (2,0)$	$\delta(2,\varepsilon) := (s,0)$
$\delta(0,1) := (3,0)$	$\delta(3,\varepsilon) := (s,1)$

Beispiel: nicht-deterministischer Automat

Akzeptiere alle Wörter der Form abababab… oder acacac….



Beispiel: deterministische Variante



Bemerkung:

Der deterministische Automat benötigt mehr Zustände.

Satz: Zu jedem nicht-deterministischen Automaten gibt es einen deterministischen Automaten, welcher dieselbe Sprache akzeptiert.

Formale Sprache

Ein **Alphabet** A ist eine nicht-leere, endliche Menge von Zeichen.

Ein **Wort** W über dem Alphabet A ist eine endliche Sequenz von Zeichen aus A.

Spezialfall: leeres Wort ε

A* ist die **Menge aller Wörter** über A (inkl. dem leeren Wort).

Eine Teilmenge L von A* heisst **Sprache** über A

Operationen

Seien u, w Wörter. Dann ist $u \circ w$ die Konkatenation von u und w. (Hintereinanderschreiben)

Seien L und M Sprachen. Dann ist

$L \cup M$ die Vereinigung von L und M,

$L \circ M$ die Menge der Konkatenationen von L und M,

d.h.: $L \circ M = \{u \circ w : u \in L \text{ und } w \in M\}$,

L^* die Menge der Konkatenationen von Wörtern aus L zusammen mit dem leeren Wort ε (Kleene Stern).

Operationen: Beispiele

Seien L={aa,b} und M={c,dd}.

Dann gilt:

$L \cup M = \{aa,b,c,dd\}$

$L^* = \{\varepsilon,aa,b,aaaa,aab,baa,bb,aaaaaa,aaaab,aabaa,\ldots\}$

$L \circ M = \{aac,aadd,bc,bdd\}$

Reguläre Sprachen

Die Klasse R **der regulären Sprachen** über dem Alphabet A ist definiert durch:

1. $\emptyset \in R$ (leere Menge)
2. $\{a\} \in R$, für $a \in A$
3. Falls $s \in R$ und $t \in R$, dann auch
   $s \cup t \in R$, $s \circ t \in R$ und $s^* \in R$

Beispiele für reguläre Sprachen:

{a,b}

{ε, ab,abab,ababab,...}

{ab,abab,ababab,...}

{ab,ac,abb,acc,abbb,accc,...}

{ab,ac,abb,abc,acb,acc,abbb,abbc,...}

Reguläre Ausdrücke

Die Klasse RA der **reguläre Ausdrücke** über dem Alphabet A ist definiert durch:

1. Ø ∈ RA und a ∈ RA für a ∈ A
2. Falls s ∈ RA und t ∈ RA, dann auch
   (s|t) ∈ RA, (st) ∈ RA und (s*) ∈ RA

Beispiele für reguläre Ausdrücke, z.T. ohne Klammern:

a|b

(ab)*

ab(ab)*

a((bb*)|(cc*))

a(b|c)(b|c)*

Theorem von Kleene

Satz:

Eine Sprache ist regulär genau dann, wenn sie von einem endlichen Automaten akzeptiert wird.

Beweisskizze: Tafel

Abschlusseigenschaften

Satz:
Seien L und M reguläre Sprachen über A.
Dann sind auch

1. $A^* - L = \{x \in A^* : \text{nicht } x \in L\}$ und
2. $L \cap M$

reguläre Sprachen

Beweis: Tafel

Grammatik

Eine **Grammatik** ist ein 4-Tupel (N,T,S,P) mit

N: endliche Menge von non-terminal Symbolen

T: endliche Menge von terminal Symbolen

S $\in$ N: Startsymbol

P: endliche Menge von Produktionen

Eine **Produktion** hat die Form $u \rightarrow v$ wobei $u,v \in (N \cup T)^*$
und u muss ein Symbol aus N enthalten.

Es gibt mindestens eine Produktion in P mit u=S.

Sprache einer Grammatik

Sei Grammatik G= (N,T,S,P).

Seien W, W‘ $\epsilon(N \cup T)^*$, W=uvw, W‘=uv‘w und $v \rightarrow v‘$,
So schreiben wir $W \triangleright W‘$.

Falls $W1 \triangleright W2 \triangleright \ldots \triangleright Wn$, so sagen Wn ist aus W1 **hergeleitet**. Die Menge aller Wörter von T, welche aus S hergeleitet werden können, ist die **Sprache**, welche durch die Grammatik G erzeugt wird.

Grammatik: Beispiel

Sei N = {S,E,N,D}, T = {0,1,(,),+,-,*,/} und P besteht aus:

S→E, D →0, D →1, N→D, N→ND,

E→N, E →(E+E), E →(E-E), E →(E*E), E →(E/E).

Folgendes sind Herleitungen aus S:

- S ▷ E ▷ N ▷ ND ▷ NDD ▷ DDD ▷ 0DD ▷ 00D ▷ 001
- S ▷ E ▷ (E+E) ▷ (N+E) ▷ (N+N) ▷ (D+N) ▷ (D+ND) ▷ (1+ND) ▷ (1+DD) ▷ (1+1D) ▷ (1+10)
- S ▷ E ▷ (E+E) ▷ (E+(E/E)) ▷ (N+(E/E)) ▷ (N+(N/E)) ▷ (N+(N/N)) ▷ **...** ▷ (D+(D/D)) ▷ (1+(D/D)) ▷ (1+(1/D)) ▷ (1+(1/0))

Reguläre Grammatik

Eine Grammatik heisst **regulär**, falls alle Produktionen die Form $n \rightarrow w$ haben, wobei:

n: non-terminal Symbol

w: ist das leere Wort oder ein Wort mit höchstens einem non-terminal Zeichen, welches am Schluss stehen muss.

Satz:

Jede reguläre Grammatik erzeugt eine reguläre Sprache.

Zu jeder regulären Sprache gibt es eine reguläre Grammatik, welche sie erzeugt.

Probleme

Sei L(G) die Sprache, welche von einer Grammatik G erzeugt wird.

- **Wortproblem**:
  Ist w in L(G) für gegebenes Wort w?
- **Leerheitsproblem**:
  Ist L(G) leer?
- **Äquivalenzproblem**:
  Ist L(G1) = L(G2) für zwei Grammatiken G1 und G2?

Vorschau

- Chomsky Hierarchie
- Kontextfreie Sprachen
- Stack-Automaten
- Turing Maschinen
- Entscheidbarkeit

- Vorlesung: Automaten und formale Sprachen
- Vorlesung: Berechenbarkeit und Komplexität

Fragen