Computational Logic

Diskrete Mathematik

Martin Avanzini   Arne Dür   Christoph Kollreider   Georg Moser

Fakultät für Mathematik, Informatik und Physik @ UIBK

Sommersemester 2010

Zusammenfassung der letzten LV

Satz — Kontraposition des Pumpinglemma

Angenommen

1. $\forall\ n$
2. $\exists$ ein String $w \in L$ mit $\ell(w) \geqslant n$ und
3. $\forall$ Zerlegung von w in x, y und z mit $w = xyz$
   - $y \neq \epsilon$,
   - $\ell(xy) \leqslant n$,
4. $\exists\ k$ mit $x(y)^k z \notin L$

dann ist L nicht regulär.

Beispiel

die Sprache L der Wörter, die genauso viele 0en wie 1er enthalten ist nicht regulär

Übersicht

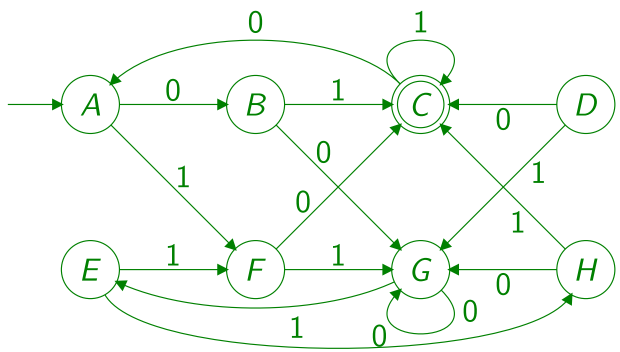
Automaten, reguläre Sprachen und Grammatiken, (nicht)-deterministische endliche Automaten, Teilmengenkonstruktion, Automaten mit ϵ-Übergängen, Umwandlung endlicher Automaten in reguläre Ausdrücke, Algebraische Gesetze für reguläre Ausdrücke, Pumpinglemma, Minimierung

Einführung in die Berechenbarkeitstheorie, Turing Maschinen, Äquivalente Formulierungen, Entscheidungsprobleme, Universelle Maschinen und Diagonalisierung,

Einführung in die Komplexitätstheorie, Laufzeitkomplexität, die Klassen P und NP, logarithmisch platzbeschränkte Reduktionen, Speicherplatzkomplexität

Äquivalenz von Zuständen

betrachte DEA ①:



Frage

Kann dieser Automat vereinfacht werden?

Antwort

Ja, denn intuitiv sind die folgenden Zustandspaare äquivalent

$$(A, E) \quad (B, H) \quad (D, F)$$

Zustandsäquivalenz

sei A ein DEA $(Q, \Sigma, \delta, q_0, F)$

Definition — äquivalent

Zustände p und q heißen äquivalent wenn

- $\forall$ Strings w
  $\widehat{\delta}(p, w)$ genau dann akzeptierend, wenn $\widehat{\delta}(q, w)$ akzeptierend

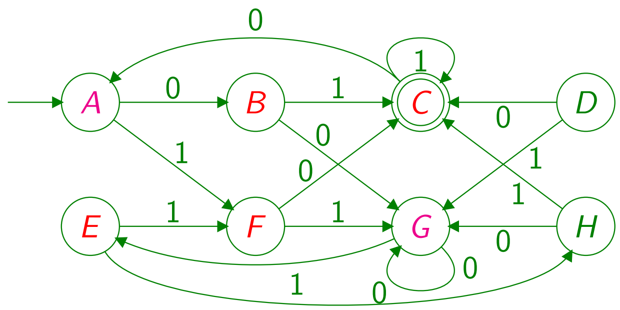
Definition — unterscheidbar (direkt)

Zustände p und q heißen unterscheidbar wenn

- $\exists$ String w
  $\widehat{\delta}(p, w)$ akzeptierend, $\widehat{\delta}(q, w)$ nicht akzeptierend (oder umgekehrt)

Beispiel

betrachte Automaten ①



betrachte A und G:

- ϵ, 0 ,1 unterscheiden nicht
- aber $\widehat{\delta}(A, 01) = C \in F$
- $\widehat{\delta}(G, 01) = E \notin F$

sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DEA

Definition — unterscheidbar (induktiv)

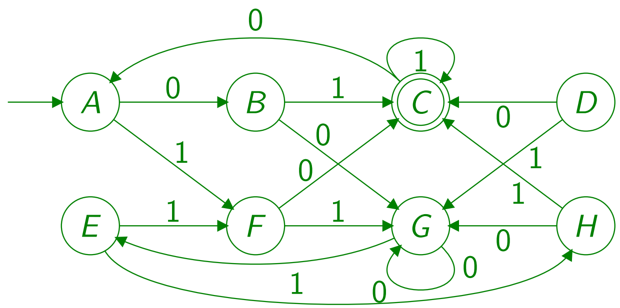
1. Basis
   wenn p akzeptierend und q nicht,
   dann sind $\{p, q\}$ unterscheidbar
2. Schritt
   seien p und q Zustände, a Eingabe und
   - sei $\delta(p, a) = r$, $\delta(q, a) = s$
   - sodass $\{r, s\}$ unterscheidbar

   dann sind $\{p, q\}$ unterscheidbar

Satz

sind Zustände p, q nach der induktiven Definition nicht unterscheidbar, dann sind sie auch äquivalent

Beispiel



	A	B	C	D	E	F	G
B	✓						
C	✓	✓					
D	✓	✓	✓				
E		✓	✓	✓			
F	✓	✓	✓		✓		
G	✓	✓	✓	✓	✓	✓	
H	✓		✓	✓	✓	✓	✓

Table-filling Algorithmus liefert die folgenden äquivalenten Zuständsblöcke $\{A, E\}$, $\{B, H\}$, $\{D, F\}$, $\{C\}$, $\{G\}$

Satz

die Äquivalenz von Zuständen ist transitiv: wenn (i) Zustände p und q äquivalent und (ii) Zustände q und r äquivalent
dann sind auch p und r äquivalent

Beweis

- $A = (Q, \Sigma, \delta, q_0, F)$
- angenommen $\{p, q\}$ und $\{q, r\}$ sind äquivalent, aber $\{p, r\}$ sind unterscheidbar
- $\exists$ String w

$$\widehat{\delta}(p, w) \in F \qquad \widehat{\delta}(r, w) \notin F$$

- also $\widehat{\delta}(q, w) \notin F$ sonst wären q und r unterscheidbar
- also $\widehat{\delta}(p, w) \in F$ und $\widehat{\delta}(q, w) \notin F$,
- somit sind p und q unterscheidbar: Widerspruch ■

Satz

die Blöcke der äquivalenten Zustände formen eine Partition

Minimierung Endlicher Automaten

Definition

Minimierungsalgorithmus

DEA $A = (Q, \Sigma, \delta, q_0, F)$
konstruiere $B = (Q_B, \Sigma, \delta_B, q_B, F_B)$:

1. unerreichbare Zustände eliminieren
2. partitioniere die Zustände in äquivalente Blöcke
3. Q_B sind die verschiedenen Blöcke
4. $\forall$ S ein Block, $\forall$ a ein Eingabesymbol:
   $\exists$ Block T, sodass für alle $q \in S$ gilt $\delta(q, a) \in T$
   setze $\delta_B(S, a) = T$
5. q_B ist der Block, der q_0 enthält
6. F_B ist die Menge der Blöcke, die Zustände aus F enthalten

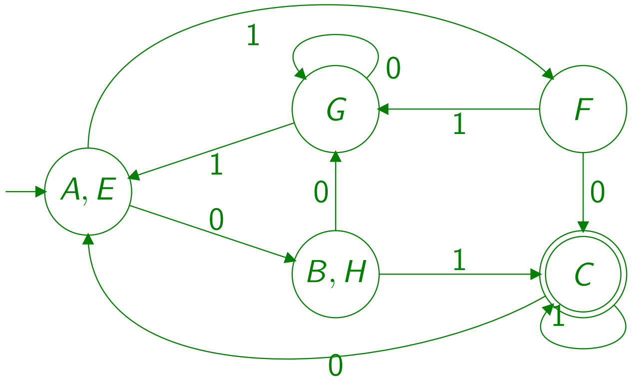
Beispiel

Minimierung des Automaten ①:

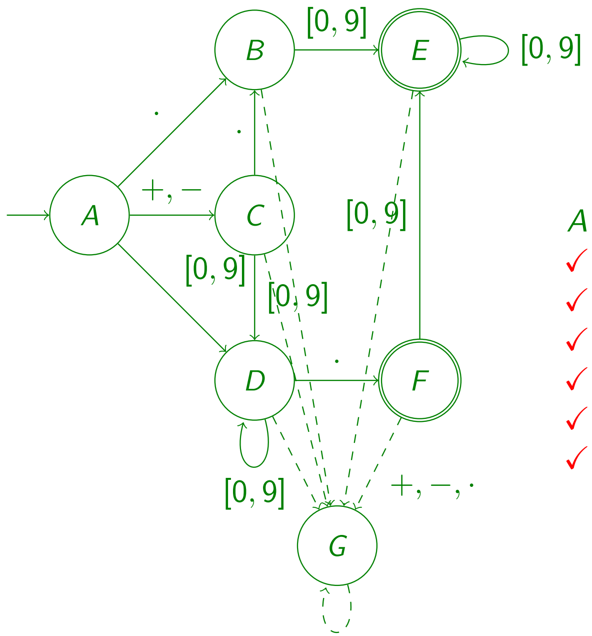
1. alle Zustände sind erreichbar
2. Zustände des minimierten Automaten ②:

$$\{A, E\} \quad \{B, H\} \quad \{C\} \quad \{F\} \quad \{G\}$$

3. Übergangsfunktion δ_2:
   - $\delta_2(\{A, E\}, 0) = \{B, H\}$, da
   - $\delta_1(A, 0) = B$
4. Ergebnis:

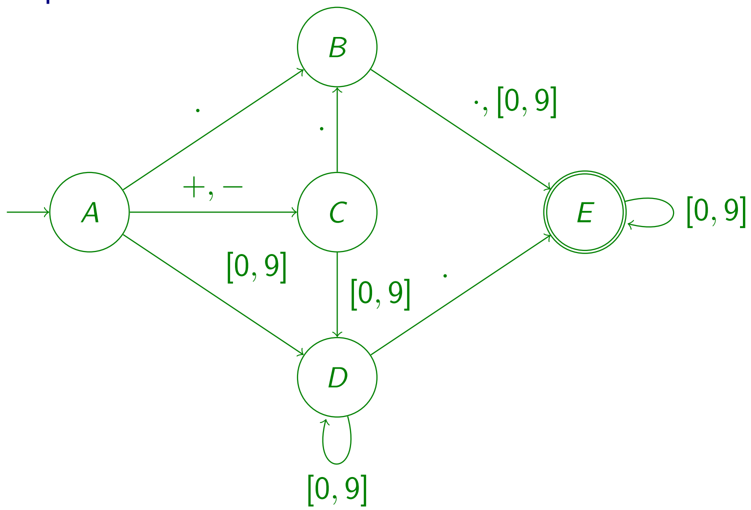


Beispiel: Der Dezimalzahlautomat



A						
✓	B					
✓	✓	C				
✓	✓	✓	D			
✓	✓	✓	✓	E		
✓	✓	✓	✓		F	
✓	✓	✓	✓	✓	✓	G

Beispiel

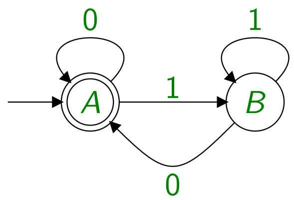


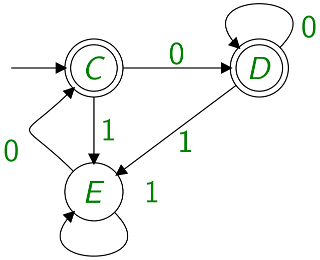
Satz

1. der Minimierungsalgorithmus ist optimal
2. das Ergebnis des Minimierungsalgorithmus ist eindeutig

Äquivalenz von Automaten

Beispiel





Beobachtung

- beide DEAs akzeptieren $L(\epsilon + (\mathbf{0} + \mathbf{1})^*\mathbf{0})$
- zur Kontrolle wende Table-filling Algorithmus an
- auf einen Automat mit zwei Startzuständen:

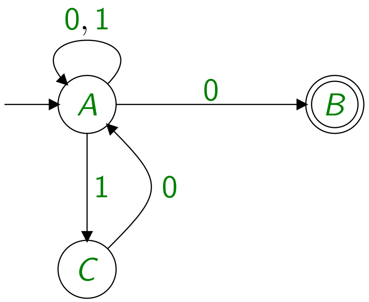
A				
✓	B			
	✓	C		
	✓		D	
✓		✓	✓	E

Minimierung eines NEA

Satz

Minimierungsalgorithmus ist nicht richtig für NEAs

Beweis



- Zustände $\{A, B\}$ sowie $\{C, B\}$ sind unterscheidbar
- ebenso ist $\{A, C\}$ unterscheidbar bei Eingabe 0
- aber Zustand C ist überflüssig

∎