

Lemma 3.32 ~~Beweis~~

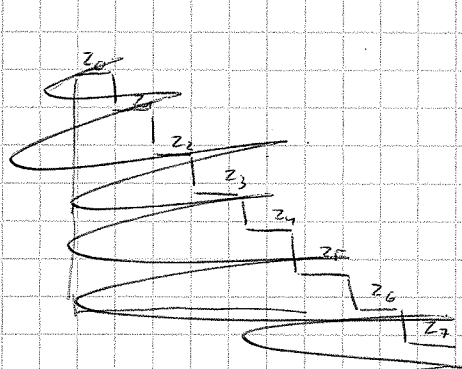
Aussage der PL: $\exists n \in \mathbb{N}^+ \text{ mit Eigenschaft } (*)$. Negation der Aussage PL:

$$\neg \exists n \in \mathbb{N}^+ \text{ mit } (*) \equiv \neg \exists n \in \mathbb{N}^+ \forall w \in L \text{ mit } |w| \geq n \exists \text{ Zerlegung } w = xyz \text{ mit } |xy| \leq n \text{ und } y \neq \epsilon \\ \equiv \forall n \in \mathbb{N}^+ \exists w \in L \text{ mit } |w| \geq n \forall \text{ Zerlegung } w = xyz \text{ mit } |xy| \leq n \text{ und } y \neq \epsilon \\ \exists i \in \mathbb{N} [xy^i z \neq w]$$

Beispiel 3.33 ($L_n \notin REG$)

- (1) Sei $n \in \mathbb{N}^+$ beliebige natürliche Zahl.
- (2) Wir definieren $w = 0^n 1^n$ ($w \in L$, $|w| \geq n$)
- (3) Sei $w = xyz$ eine beliebige Zerlegung von w mit $y \neq \epsilon$ und $|xy| \leq n$
- (4) Wir definieren $i = 0$ und zeigen: $xy^i z \notin L$
 - Aus $|xy| \leq n$ folgt $x = 0^i$, $y = 0^k$ für $i, k \in \mathbb{N}$
 - Aus $y \neq \epsilon$ folgt $k \geq 1$
 - $\Rightarrow w = xyz = \underbrace{0^i}_x \underbrace{0^k}_y \underbrace{0^{n-i-k} 1^n}_z$
 - $\Rightarrow xy^0 z = 0^i 0^{n-i-k} 1^n = 0^{n-k} 1^n$ mit $k \geq 1 \notin L_n$
 - $\Rightarrow L_n$ erfüllt nicht die PL $\Rightarrow L_n \notin REG$ $\square$

Beispiel 3.40 (vgl. Bsp. 3.34)



z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X

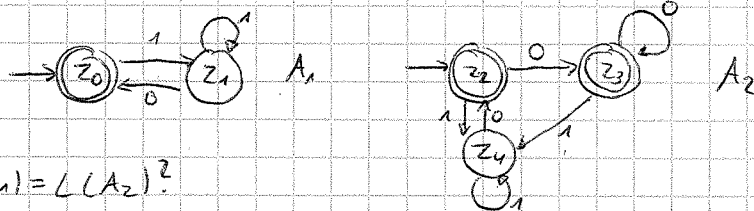
R_1
 R_2
 R_3

Nicht markierte Paare sind nicht unterschiedbar und damit äquivalent

$\{z_0, z_1\} \rightarrow \{z_2, z_5\}$ schon markiert

Beispiel 3.42 Äquivalenztest bei DEAs

gegeben



Frage $L(A_1) = L(A_2)$?

Wir bestimmen unterschiedbare Zustandspaare

z_0	z_1	z_2	z_3	z_4
X	X	X	X	X
X	X	X	X	X
X	X	X	X	X
X	X	X	X	X

R_1, R_2

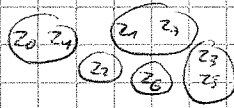
$\rightarrow$ Wir sehen: $\{z_0, z_2\}$ sind nicht unterschiedbar und damit äquivalent, also akzeptieren beide dieselbe Sprache $L(A_1) = L(A_2)$

Beispiel 3.44 Partitionierung der Zustandsmenge 3.54

$\{z_0, z_4\}, \{z_3, z_5\}, \{z_1, z_7\}$ äquivalent

Zustandsmenge $Z = \{z_0 \dots z_7\}$ part. wie folgt:

$$[z_0] = [z_4] = \{z_0, z_4\}; [z_1] = [z_7] = \{z_1, z_7\}, [z_2] = \{z_2\}, [z_3] = [z_5] = \{z_3, z_5\}, \\ [z_6] = \{z_6\}$$



Es fehlen Bsp 3.34, 3.46, Satz 3.47 (3.6.15)

zu Def. 4.6 (Typen von Grammatiken)

typische Regeln:

Typ 0 $abAAaC \rightarrow XaAca$

1 $aBabAcBc \rightarrow aBabBcBcBc$

2 $A \rightarrow cAcBc$

3 $A \rightarrow aB$ oder $A \rightarrow c$

Kontextsensitiv

Kontextfrei

regulär

Satz 4.11 $L \subseteq \Sigma^*$: L regulär $\Leftrightarrow L$ ist vom Typ 3

Beweis

- „1 $\Rightarrow$ 2“: Sei $L = L(A)$ mit NEA $A = (\Sigma, Z, \delta, z_0, F)$. Konstruiere Typ-3-Grammatik G .

Idee: $\begin{matrix} \textcircled{z} & \xrightarrow{a} & \textcircled{z'} \\ \textcircled{z} & \xrightarrow{a} & \textcircled{z} \end{matrix}$ wird simuliert durch $z \rightarrow az'$
 $\begin{matrix} \textcircled{z} & \xrightarrow{a} & \textcircled{z} \end{matrix}$ wird simuliert durch $z \rightarrow a$ (kein NT mehr übrig)

$G =_{\text{def}} (\Sigma, Z, z_0, R)$ mit $R := \{z \rightarrow az' \mid z, z' \in Z, a \in \Sigma, z' \in \delta(z, a)\} \cup \{z \rightarrow a \mid z \in Z, a \in \Sigma, \delta(z, a) \cap F \neq \emptyset\}$

Sei $w = a_1 \dots a_n$ mit $n \geq 1$ und $a_1 \dots a_n \in \Sigma$. $w \in L(A) \Leftrightarrow$ es ex. $s_1 \dots s_n$ mit $s_i \in F$ und $z_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s_n \Leftrightarrow \dots$ sodass

$z_0 \rightarrow a_1 s_1, s_1 \rightarrow a_2 s_2, \dots, s_{n-1} \rightarrow a_n$ Regeln aus R sind

$\Leftrightarrow z_0 \xrightarrow{*} a_1 \dots a_n \Leftrightarrow w \in L(G)$

Damit gilt: $L(A) = L(G) \cup \{ \epsilon \}$

- „2 $\Rightarrow$ 1“: Sei L vom Typ 3 also $L = L(G)$ oder $L = L(G) \cup \{ \epsilon \}$ mit G vom Typ 3

g.z.z. $L(G)$ regulär (weil $\{ \epsilon \}$ regulär und REG unter $\cup$ abgeschlossen)

Bau NEA A für $L(G)$, Idee: $A \rightarrow aB$ werden simuliert durch $\textcircled{A} \xrightarrow{a} \textcircled{B} \dots$

$A \rightarrow a$ werden simuliert durch $\textcircled{A} \xrightarrow{a} \textcircled{\times}$

$A = (\Sigma, N \cup \{X\}, \delta, S, \{X\})$, wobei X ein neues Symbol ist

$\delta(A, a) := \{B \mid A \rightarrow aB \in R\}$ falls $A \rightarrow a$ keine Regel in R ist

falls $A = X$

$\{ \epsilon \} \cup \{X\}$ falls $A \rightarrow a$ eine Regel in R ist

Sei $w = a_1 \dots a_n$ mit $n \geq 1, a_1 \dots a_n \in \Sigma$. $w \in L(G) \Leftrightarrow S \xrightarrow{*} a_1 \dots a_n$

$\Leftrightarrow$ es ex. $A_2 \dots A_n \in N$ mit $S \rightarrow a_1 A_2, A_2 \rightarrow a_2 A_3, \dots, A_n \rightarrow a_n$ sind Regeln

$\Leftrightarrow$ es ex. $A_2 \dots A_n \in N$ mit $\textcircled{S} \xrightarrow{a_1} \textcircled{A_2} \xrightarrow{a_2} \dots \xrightarrow{a_n} \textcircled{\times} \Leftrightarrow a_1 \dots a_n \in L(A)$

$\Rightarrow L(G)$ regulär



Satz 4.13 ϵ in kontextfreier Grammatik ist nicht

Beweis

- „1 $\Rightarrow$ 2“: Sei $G = (\Sigma, N, S, R)$ kontextfreie Grammatik mit $L(G) = L - \{ \epsilon \}$

• $\epsilon \notin L$, also $L(G) = L$ und fertig

• $\epsilon \in L$: neues Nicht-Terminal T zur Grammatik als Startsymbol und die neuen Regeln $T \rightarrow S, T \rightarrow \epsilon$

- „2 $\Rightarrow$ 1“ Sei G eine Grammatik wie in 2 beschrieben. Wir erweitern R wie folgt:

Ist $A \rightarrow u_0 B_1 u_1 \dots B_k u_{k+1}$ eine Regel mit $u_i \in (\Sigma \cup N)^*$, $B_i \in N$ und $u_i \neq \epsilon$

$B_i \xrightarrow{*} \epsilon$ für $1 \leq i \leq k$, so nehmen folgende Regel hinzu:

$A \rightarrow u_0 u_1 \dots u_{k+1}$ (Die so gewonnene Regelmenge R' mit (*) kann auf die

ϵ -Regeln verzichten und wir erhalten $L(G) - \{ \epsilon \} = L(G')$, wobei

$G' = (\Sigma, N, S, R' - \{ A \rightarrow \epsilon \mid A \in N \})$ kontextfrei ist.

$\Rightarrow L(G)$ kontextfrei



z.B. führt $A \rightarrow aBbCc$ mit $B \xrightarrow{*} \epsilon$ und $C \xrightarrow{*} \epsilon$ zu folgenden neuen Regeln
 $A \rightarrow abc$, $A \rightarrow AbBc$, $A \rightarrow aBbCc$, $A \rightarrow aBbBc$, $A \rightarrow aBbC$, $A \rightarrow abBc$,
 $A \rightarrow abCc$

Beispiel 4.16 (Umwandlung in Chomsky-Normalform)

$G = (\Sigma, Z, S, R)$ mit $\Sigma = \{a, b\}$ und $Z = \{S\}$

$R = \{S \rightarrow aSa, S \rightarrow bSb, S \rightarrow aa, S \rightarrow bb, S \rightarrow a, S \rightarrow b\}$

Σ N
 kein ans

1. Terminale auf rechte Seite verschieben

- $D_a, D_b \in Z$ werden gewählt
- neue Regeln: $D_a \rightarrow a, D_b \rightarrow b$
- alle Regeln: a durch D_a ersetzen

Danach: $G_1 = (\{a, b\}, \{S, D_a, D_b\}, S, R_1)$ mit

$R_1 = \{S \rightarrow D_a S D_a, S \rightarrow D_b S D_b, S \rightarrow D_a D_a, S \rightarrow D_b D_b, S \rightarrow D_a, S \rightarrow D_b, D_a \rightarrow a, D_b \rightarrow b\}$

2. Regeln der Form $A \rightarrow B$ beseitigen

- $S \rightarrow a$, weil $S \xrightarrow{*} D_a \xrightarrow{*} a$ (und $S \rightarrow b$) hinzufügen
- eliminiere $S \rightarrow D_a$ (und $S \rightarrow D_b$)

Danach: $G_2 = (\{a, b\}, \{S, D_a, D_b\}, S, R_2)$ mit

R_2 wie oben, ohne $S \rightarrow D_a, S \rightarrow D_b$, mit $S \rightarrow a, S \rightarrow b$

3. Regeln der Form $A \rightarrow B_1 \dots B_k, k \geq 3$

- ersetze $S \rightarrow D_a S D_a$ durch $S \rightarrow D_a E_1$ und $E_1 \rightarrow S D_a$
- ersetze $S \rightarrow D_b S D_b$ durch $S \rightarrow D_b E_2$ und $E_2 \rightarrow S D_b$

$G_3 = (\{a, b\}, \{S, D_a, D_b, E_1, E_2\}, S, R_3)$, wobei
 $R_3 = \{ \dots \}$

$\Rightarrow G_3$ ist in Chomsky-NF und $L(G_3) = L(G)$

11.6.15

Lemma 4.17 (Pumping-Lemma für kontextfreie Sprachen)

Beweis:

- Sei L kontextfrei o.B.d.A. $\epsilon \in L$ (sonst folgt $L = \{\epsilon\}$ und steht fest, dass sich die Aussage ergibt)

Nach Satz 4.15 ex. Grammatik $G = (\Sigma, N, S, R)$ in CNF, die unsere Sprache erzeugt.

- $n = 2^{|N|}$

- Sei $z \in L$ mit $|z| \geq n$

- Betrachten den Erzeugungsbaum von z , es ist ein Binärbaum (wegen CNF), z.B.

Dort ex. ein Weg von Wurzel zu einem Blatt, auf dem mind. $|N|+1$ Nichtterminale durchlaufen

werden. Beweis: angenommen alle Wege besitzen $\leq |N|$ Nichtterminale

- wegen Binärbaum-Eigenschaft ist die Länge des erzeugten Wortes
 $\leq 2^{|N|-1}$, d.h. $|z| < n$ Widerspruch zu Wahl von z

- Sei $(A_0, A_1, \dots, A_{|N|}, a)$ der linke Abschnitt eines längsten Weges im Baum, wobei

$A_i \in N, a \in \Sigma$; Schlussabschluss: auf dem Weg taucht mindestens ein Nichtterminal ϵ doppelt auf, d.h. $\exists i, j$ mit $0 \leq i, j \leq |N|$ und $A_i = A_j$

- Situationen im Erzeugungsbaum:

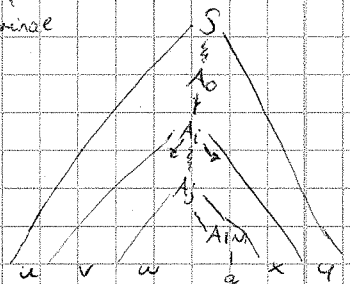
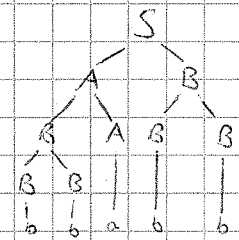
Weg von A_i zu Blättern enthalten max $|N|$ Nichtterminale (wobei wir längsten Weg gewählt hatten) $\Rightarrow |vwx| \leq 2^{|N|} = n$

auf A_i wird die Regel $A_i \rightarrow BC$ angewendet, d.h. in vwx taucht das von B erzeugte oder das von C erzeugte auf. (Beide nicht leer, da keine ϵ -Regel), folglich $|vwx| \geq 1 \Rightarrow vwx \neq \epsilon$

Wegen $A_i = A_j$ kann:

- Baum unter A_i an A_j angefügt werden $\Rightarrow$ ergibt $uv^iwx^i y \in L$
- Baum unter A_i weggelassen anstelle von A_j angefügt werden $\Rightarrow uv^iwx^i y \in L$

□



Beispiel 4.18 (Anwendung des Pumping-Lemmas)

$L = \{a^n b^n c^n \mid n \geq 1\}$ nicht kontextfrei

Beweis durch Widerspruch:

Angenommen L kontextfrei. Dann gibt für L das PLKS also ex. $n \in \mathbb{N}^+$ mit (*)

Wir erhalten Widerspruch, indem wir (*) für festes n prüfen: $z = a^n b^n c^n \in L, |z| \geq n$

Aus (*) folgt: ex. Zerlegung $z = uvwx$ sodass $|vwx| \leq n$ und $|vwx| \leq n$ und $|vwx| \leq n$ und $|vwx| \leq n$

Sei $z = uvwx$ die beliebige Zerlegung

1. Fall: In v oder x kommen verschiedene Buchstaben vor

$\Rightarrow uv^2wx^2y \notin a^n b^n c^n$ und damit $\notin L$ $\nmid$ zu PL

2. Fall: v und x bestehen jeweils aus gleichen Buchstaben

$\Rightarrow$ es ex. ein Buchstabe, der weder in v noch in x vorkommt; dieser kommt in uvw n -mal vor. (1)

$\Rightarrow |uvw| \leq |z| = 3n$ (weil $|vwx| \geq 1$) (2)

Aus (1) und (2) folgt, dass die Anzahl der a, b, c in uvw nicht gleich ist, also

$uvw \notin L$ $\nmid$ zu PL

$\Rightarrow L$ nicht kontextfrei $\square$

Satz 4.21 (Abschlussregeln von L_2)

Beweis: 1. Vereinigung: zwei Grammatiken, Mengen vereinigen, neue Startsymbol

2. a, $T_1 = \{a^n b^n c^m \mid m, n \geq 1\} \in L_2$, da von folgender kontextfreier Grammatik erzeugt.

$S \rightarrow S' C, S' \rightarrow a S' b, S' \rightarrow ab, C \rightarrow c C, C \rightarrow c$

$T_2 = \{a^k b^m c^n \mid m, n \geq 1\} \in L_2$ (analog)

$T_1 \cap T_2 = \{a^k b^m c^n \mid k=m \text{ und } m=n\} = \{a^k b^k c^k \mid k \geq 1\} \Rightarrow \text{Bsp. 4.18}$

$\Rightarrow$ nicht kontextfrei

$\Rightarrow$ kein Abschluss unter $\cap$

b, ans. welche unter Kompl. abgeschlossen:

$A_1, A_2 \in L_2 \Rightarrow \overline{A_1 \cup A_2} \in L_2$, weil L_2 abgeschlossen
 $= A_1 \cap A_2$ $\nmid$ 2-a

// 17.6.15: nicht anwesend: siehe Facebook-Upload von Christian //

18.6.15 Wiederholung: - CYK-Algo: Entscheidbarkeit kontextfreier Sprachen in Zeit $O(n^3)$

- nicht-deterministische Kellerautomaten

- Satz: $L \in L_2 \Leftrightarrow L$ von n.d. KA akzeptiert

- Satz: L wird von kontextsensitiver Grammatik erzeugt $\Leftrightarrow L$ wird von nichtreduzierenden G. erzeugt

- Linear beschränkte Automaten $\exists a|b|c|d$

zum Beweis von Satz 4.32, 1 & 2:

Definieren nichtreduzierende Grammatik $G = (Z, M, S, R)$ für L :

1. Schritt: Definition einer Hilfsregelmenge R' , die auf Konfigurationen die Ableit. von H' simuliert.

Bsp: $(z', b, L) \in f(z, a) \Rightarrow c(z, a) \rightarrow (z', c) b \in R'$ für alle $c \in \Gamma$

$(z', b, R) \in f(z, a) \Rightarrow (z, a) c \rightarrow b(z', c) \in R'$ für alle $c \in \Gamma$

$(z', b, O) \in f(z, a) \Rightarrow (z, a) \rightarrow (z', b) \in R'$

Für alle Konfigs K, K' von M' gilt dann: Es gibt eine Rechnung, die von K nach K' führt $\Leftrightarrow K \xrightarrow{R'} K'$ mit Hilfe der Regeln aus R'

2. Schritt: $N := \{S, A\} \cup (\Delta \times \Sigma)$, wobei $\Delta = \Gamma \cup (Z \times \Gamma)$ (Alphabet für Konfigs $\neq$ von M')

3. Schritt: Definiere die Regeln von R :

1, $S \rightarrow A(a, a)$ für $a \in \Sigma$ | Diese Regeln erzeugen folgende Wörter:

$A \rightarrow A(a_1, a_1)$ "

$A \rightarrow (z_0, a_1) a_1$ "

"

"

"

"

"

"

"

"

"

$(z_0, a_1) a_1 a_2 a_3 \dots a_n$

$a_1 a_2 a_3 \dots a_n$

$a_1 a_2 a_3 \dots a_n$

$a_1 a_2 a_3 \dots a_n$

$a_1 a_2 a_3 \dots a_n$

$a_1 a_2 a_3 \dots a_n$

$a_1 a_2 a_3 \dots a_n$

$a_1 a_2 a_3 \dots a_n$

$a_1 a_2 a_3 \dots a_n$

oben wird Berechnung des LBA simuliert
unten wird Eingabe gepulst

2., $(A_1, a)(A_2, b) \rightarrow (B_1, a)(B_2, b)$ für $a, b \in \Sigma, A_1, A_2, B_1, B_2 \in \Delta, A_1 A_2 \rightarrow B_1 B_2 \in R'$

$(A, a) \rightarrow (B, a)$ für $a \in \Sigma, A, B \in \Delta, A \rightarrow B \in R'$

Diese Regeln simulieren Rechnung von M' in der ersten Komponente und lassen die zweite Komponente unverändert

$$3. ((2, a, b) \rightarrow b \quad \text{für } a \in \Gamma, b \in \Sigma \\ (a, b) \rightarrow b$$

Hier entsteht nur dann ein Wort über Sigma, wenn der Stoppzustand z_1 erreicht wurde.

Diese Regeln bilden nach einem Schritt die 1. Komponente der Rückstufen:

Üb: $\text{Reit } a_1 \dots a_n \in \Sigma^*$

Mit den obigen Regelungen gilt für $a_1 \dots a_n \in \Sigma^*$:

$a_1 \dots a_n \in L \Leftrightarrow$ ~~es~~ ^{besteht} auf $a_1 \dots a_n$ ~~ein~~ ^{mindestens} eine abz. Reduzierung

$\Leftrightarrow M'$ auf $a_1 \dots a_n$ besitzt eine abz. Reduzierung, die stoppt

$\Leftrightarrow S \xrightarrow{G} a_1 \dots a_n \Rightarrow L = L(G)$

weil G nicht-reduzierende Grammatik ist $\Rightarrow$ ^{4.30} L ist kontextsensitiv



Folgerung 4.33 Jede kontextsensitive Sprache ist in $2^{O(n)}$ entscheidbar.

Beweis: nach 4.32 genügt es zu zeigen, dass jede von einem LBA $M = (\Sigma, \Gamma, \delta, z_0, z_1)$ akzeptierte Sprache in Zeit $2^{O(n)}$ entscheidbar sind

Bestimmen die Anzahl der Bits mit denen sich Konfigurationen von M bei Eingaben der Länge n beschreiben lassen:

$$b(n) := \underbrace{\lceil \log(|\Sigma|) \rceil \cdot (n+2)}_{\text{zum Speichern des Bandinhalts (n+2) Zellen, die jeweils ein Symbol aus \Sigma enthalten}} + \underbrace{\lceil \log(n+2) \rceil}_{\text{zum Speichern der Kopfposition auf dem Band}} + \underbrace{\lceil \log(|\Gamma|) \rceil}_{\text{zum Speichern des Zustands}}$$

(*) $b(n) \in O(n)$, denn $|\Sigma|$ und $|\Gamma|$ sind konstant für jeden LBA

(**) Für Eingaben der Länge n besitzt M maximal $2^{O(n)}$ Konfigurationen

Wir bestimmen nun die aus der Startkonfiguration erreichbaren Konfigurationen genauso wie für DEAs in 3.45 und testen, ob darunter eine akzeptierende ist.

1. $R = \{\text{Startkonf.}\}$
2. while ($\exists K \in R$, von der sich in einem Rechenschritt ein $K' \notin R$ erreichen lässt)
3. $R = R \cup \{K'\}$
4. if (R enthält abz. Konfig.) then return 1
5. else return 0

Laufzeit: 1. $O(n)$ zum Erstellen der Startkonfigurationen

2. max. $2^{O(n)}$ Durchläufe, da pro Durchlauf eine neue Konf. zu R hinzugefügt wird (nach (**))

3. Laufzeit pro Durchlauf $2^{O(n)} \cdot O(n) = O(n) \cdot 2^{O(n)} + O(n)$ (Anzahl der Konfig. in R) (Zeit, ob eine der Konfig. einen Nachfolger in R hat)

$= 2^{O(n)}$

4. $2^{O(n)} \cdot O(n) = 2^{O(n)}$

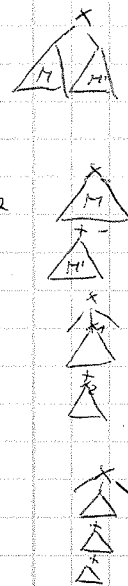
Gesamtlauftzeit: $O(n) + 2^{O(n)} + 2^{O(n)} + 2^{O(n)} = 2^{O(n)}$



Satz 4.34 (Abschlusseigenschaften von L_1)

Beweis: Seien $L, L' \in L_1$; M und M' LBAs die jeweils L, L' akzeptieren.
Nach 4.32 g.z.z. LBAs für folgende Sprachen anzugeben:
 $L \cup L'$, $L \cap L'$, $L \cdot L'$, L^* und $\bar{L}$

- " $\cup$ ": LBA für $L \cup L'$ bei Eingabe x :
 - Verzweigung in zwei Rechenwege: in 1. wird M auf x simuliert, in 2. M' auf x
- " $\cap$ ": LBA für $L \cap L'$ bei Eingabe x :
 - Wähle genügend große Alphabete, sodass M und M' simulieren kann und gleichzeitig die Eingabe speichern kann (z.B. $\Sigma \times \Sigma$)
 - LBA simuliert M auf x und auf dem akzeptierenden Rechenweg simuliert zusätzlich $M'(x)$ (dazu wird gespeicherte Eingabe benötigt)
 - LBA akzeptiert x , falls bei 2. Simulation akz. Rechenweg gefunden wird
- " $\cdot$ ": LBA für $L \cdot L'$ bei Eingabe
 - LBA rat nicht-def. alle Zerlegungen $x = x_1 x_2$ mit $x_1, x_2 \in \Sigma^*$
 - LBA simuliert $M(x_1)$ und auf akz. Rechenwegen $M'(x_2)$
 - LBA akzeptiert, falls eine Zerlegung $x = x_1 x_2$ gefunden wurde, sodass, beide akzeptieren
- " $*$ ": Wie zuvor, nur dass Zerlegungen mit mehreren Faktoren getrennt werden
 $x = x_1 \dots x_m$ mit $m \leq |x| + 1$ und dann $M(x_1)$ bis $M(x_m)$ nacheinander simuliert werden



"Korrt.": ohne Beweis



24. Juni 2015

$L \subseteq \Sigma^*$ ist aufzählbar $\Leftrightarrow L = \emptyset$ oder ex. totales Entscheidens $f: \mathbb{N} \rightarrow \Sigma^*$ mit $W_f = L$ (Ullmann-Algo.)

Satz 5.12 Abschlusseigenschaften von P und FP

- Beweis:
1. neue Funktion durch einfaches While-Programm realisiert
 2. $C_{A \cup B}(x) = \max\{C_A(x), C_B(x)\}$
 $C_{A \cap B}(x) = \min\{C_A(x), C_B(x)\}$
 $C_{\bar{A}}(x) = 1 - C_A(x)$



Beispiel: f_{log} mit $f_{\text{log}} \in FP$, $\Rightarrow$ nach 5.10 ex. p_1, p_2 mit
 • f wird in Zeit p_1 berechnet und $|f(x)| \leq p_1(|x|)$
 • $|g(x)| \leq p_2(|x|)$
 Folgendes Programm berechnet $f \circ g$:
 def $f \circ g(x)$:
 return $f(g(x))$

Laufzeit: $p_1(p_2(|x|))$ für $g(x)$
 + $p_1(p_2(|x|))$ für $f(y)$ mit $y = g(x)$
 + 1 return

Bsp. 5.15.2

25. Juni 2015

Wiederholung: L Typ 0 $\Leftrightarrow L$ aufzählbar, L_0 abgeschlossen unter $\cup, \cap, \cdot, ^*$, also nicht -
 FP, P, EXP

Satz 5.22 $P \neq EXP$

Beweis: Wir konstruieren ein $L \in EXP - P$. Sei M_j, M_{j+1} die auf S. 164 der Godelisierung der RAMs beschriebenen Arbeitsweise der RAM M bei Eingabe j :

1. $n = |j|$
2. simuliere $n^n + n$ Rechenschritte der Berechnung $M_j(j)$
3. $r =$ Inhalt von RO am Ende der Simulation
4. if $r = 0$ then return 1 else return 0

Sei $L = L(M)$

vgl. Unentscheidbarkeit des Halteproblems

Wir zeigen: $L \in EXP$

Laufzeit von M bei Eingabe j ($n = |j|$)

Zeile 1: $O(n)$ Rechenschritte

Zeile 2: - ein einzelner Rechenschritt von M_j kann in der Zeit $p(n)$ für ein Polynom p simuliert werden
 $\Rightarrow p(n) \cdot (n^n + n)$ Rechenschritte

Zeile 3, 4: 2 Rechenschritte

$\Rightarrow$ Laufzeit $2^{p(n)}$ für geeignetes Polynom q

$\Rightarrow L \in EXP$

Wir zeigen nun: $L \in P$

Angenommen: $L \in P \xrightarrow{5.21} \exists j \in \mathbb{N}$ mit (i) $L = L(M_j)$

(ii) $M_j(x)$ stoppt nach max. $|x|^{1/2} + |j|$ Rechenschritten

$j \in L \Leftrightarrow M_j(j)$ liefert 1

$\Leftrightarrow$ nach $|j|^{1/2} + |j|$ Schritten der Berechnung von $M_j(j)$ gilt $[RO] = 0$

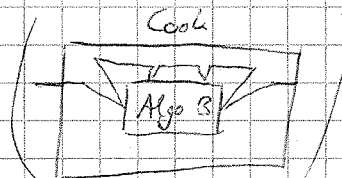
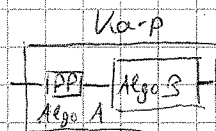
(ii) $M_j(j)$ liefert 0 (komp. Berechnung simuliert)

$\Leftrightarrow j \notin L(M_j) = L$

$\downarrow$ $L \notin P$

1. Juli 2015

zu Def. 5.28:



Satz 5.30: $SOS \in RUCKSACK$

Beweis:

$$\langle a_1, \dots, a_n, b \rangle \in SOS \Leftrightarrow \exists I \subseteq \{1, \dots, n\} \text{ mit } \sum_{i \in I} a_i = b$$

$$\Leftrightarrow \exists I \subseteq \{1, \dots, n\} \text{ mit } \sum_{i \in I} a_i \geq b \wedge \sum_{i \notin I} a_i \leq b$$

$$\Leftrightarrow \langle a_1, \dots, a_n, a_1, \dots, a_n, b, b \rangle \in RUCKSACK$$

(Gaußsche) (Cook'sche) (S) (V)

$$\Leftrightarrow f(\langle a_1, \dots, a_n, b \rangle) \in RUCKSACK, \text{ wobei}$$

$$f(\langle a_1, \dots, a_n, b \rangle) := \langle a_1, \dots, a_n, a_1, \dots, a_n, b, b \rangle, \text{ FOFP}$$

$$\Rightarrow SOS \in RUCKSACK$$



Satz 5.32 (Abgeschlossenheit von P/NP unter Reduzierbarkeit)

Beweis: 1. $B \in P$ also $C_B \in FP$,
 $A \leq B$ also ex. $f \in FP$ mit $(x \in A \Leftrightarrow f(x) \in B)$
 $\Rightarrow C_A(x) = \underbrace{C_B(f(x))}_{\in FP} \Rightarrow C_A \in FP \Rightarrow A \in P$

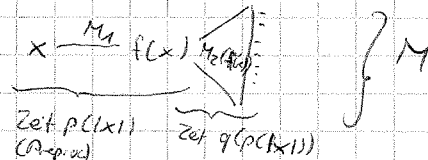
2. ~~$A \in B$~~ $A \leq B$ also ex. $f \in FP$ mit $(x \in A \Leftrightarrow f(x) \in B)$

Aus 5.10 folgt mit $f \in FP$:

$\exists$ Polynom p und eine det. TH M_1 , die f in p Zeit berechnet und $|f(x)| \leq p(|x|)$

Aus $B \in NP$ folgt, es ex. eine nicht-det. TH M_2 , die B in der Zeit q akzeptiert (q Polynom)

Konstruieren Nicht-det. TH M :



M akzeptiert $x \Leftrightarrow$

M_2 akzeptiert $f(x) \Leftrightarrow$

$\Leftrightarrow f(x) \in B$

$\Leftrightarrow x \in A$

$\Rightarrow M$ akzeptiert A in polynomieller Zeit $\Rightarrow A \in NP$

□

Satz 5.33

Beweis: 1. Wähle $a \in B$, $b \in \bar{B}$

$$f(x) = \begin{cases} a & \text{falls } x \in A \\ b & \text{sonst} \end{cases}, \quad f \in FP, \text{ weil } A \in P$$

$$\left. \begin{aligned} x \in A &\Rightarrow f(x) = a \in B \\ x \notin A &\Rightarrow f(x) = b \notin B \end{aligned} \right\} x \in A \Leftrightarrow f(x) \in B \Rightarrow A \leq B$$

2. folgt direkt aus 1.

□

für welche A gilt $A \in \emptyset$?
 für welche B gilt $B \in N$?

$P = \{N, \emptyset\}$

$N \neq \emptyset$

Ang. $A \in \emptyset$, d.h. ex. $f \in FP$ mit $x \in A \Leftrightarrow f(x) \in \emptyset \Rightarrow A = \emptyset$

Ang. $B \in N$, d.h. ex. $f \in FP$ mit $x \in B \Leftrightarrow f(x) \in N \Rightarrow B = N$

Eigenschaft 5.35

Beweis: 1. A NP-vollständig, $B \in NP \Rightarrow B \leq A \} A \equiv B$
 B NP-vollständig, $A \in NP \Rightarrow A \leq B$

2. $\Rightarrow$: Sei A NP-vollständig, also $\forall D \in NP, D \leq A$
 $A \in P \Rightarrow D \in P$ für alle $D \in NP \Rightarrow NP \subseteq P \xrightarrow{5.18} P = NP$

$\Leftarrow$: $P = NP \Rightarrow A \in NP \subseteq P \Rightarrow A \in P$

3. Sei B NP-vollständig und $C \in NP$ mit $B \leq C$; für belieiges $D \in NP$ gilt:

$D \leq B$ weil B NP-vollständig $\} D \leq C$ wegen Transitivität

$B \leq C$ nach Annahme

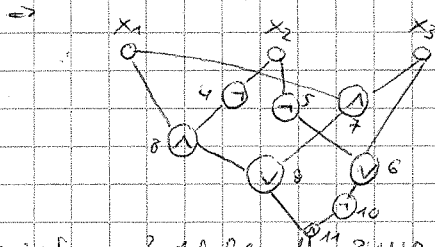
$\Rightarrow$ für alle $D \in NP$ gilt $D \leq C \} \Rightarrow C$ ist NP-vollständig $C \in NP$

□

Satz 5.45 3SAT NP-vollständig

Wir zeigen SAT $\leq$ 3SAT, aus 5.35 folgt dann 3SAT NP-complett.
Sei $F = F(x_1, \dots, x_n)$ eine beliebige boolesche Formel mit Variablen $x_1, \dots, x_n$.

- Stellen F als Schaltkreis dar, wobei die Eingänge den Variablen $x_1 \dots x_n$ entsprechen und die restlichen Knoten ~~neue~~ neuen Variablen $x_{n+1} \dots x_m$

$$F \models ((x_1 \wedge \neg x_2) \vee (x_1 \wedge x_3)) / \wedge (\neg x_2 \vee x_3)$$


= Falls $x_i = \ominus$ mit Vorgänger x_{i-1}

$$T_2 = (\neg x_c \vee \neg x_a) \wedge (x_c \vee x_a) \equiv (x_c \leftrightarrow \neg x_a) \quad (\neg x_c = \neg x_a)$$

= Falls $x_i = \textcircled{\wedge}$ ist Vorgänger x_a und x_b :

$$\begin{aligned} F_i &= (\neg x_a \vee \neg x_b \vee x_i) \wedge (\neg x_i \vee x_a) \wedge (\neg x_i \vee x_b) \\ &\equiv (x_i \leftrightarrow (x_a \wedge x_b)) \end{aligned}$$

• Fall: $x_i = \odot$ mit Vorge x_a und x_b

$$\begin{aligned} F_i &:= (\neg x_a \vee x_i) \wedge (\neg x_b \vee x_i) \wedge (\neg x_i \vee x_a \vee x_b) \\ &\equiv (x_i \leftrightarrow (x_a \vee x_b)) \end{aligned}$$

$$G_F(x_1, \dots, x_m) = \bigwedge_{C \in A_1 \cup \dots \cup A_m} F_C \quad \text{JKUF-Formeln}$$

$$H_F(x_1, \dots, x_m) := \bigwedge_{i=1}^m G_F \wedge x_i$$

Es gilt: $G_F(b_1, \dots, b_n) = 1 \Leftrightarrow b_1, \dots, b_n$ sind die in Schallkreis erzeugten Werte,
wenn ~~man~~ an den Eingängen die Werte b_1 bis b_n anliegen

Es folgt: $\langle F \rangle \in \text{SAT} \Leftrightarrow \exists b_1, \dots, b_n \in \{0, 1\}, F(b_1, \dots, b_n) = 1$

$\Leftrightarrow$ ————— sukzessive Schalthres bei Eingabe $b_n, b_{n-1}, \dots, 1$ erzeugt

($\Leftarrow$) Zb. $b_m \in \{0, 1\}$, sodass (b_{m+1}, b_m) die in Schritt 1 erzeugten Werte sind, wenn an den Eingängen b_1, b_m anliest) und am Ausgang eine 1 liegt ($b_m = 1$)

$$\Leftrightarrow \exists b_1, \dots, b_m \in \{0, 1\} \text{ s.t. } G_F(b_1, \dots, b_m) = 1 \text{ und } b_m = 1$$

(2) $\text{seclow } HF(b_n, b_m) = 1$

$$\hookrightarrow \langle H_F \rangle \in 3SAT$$

Mit $f(\langle F \rangle) = \langle H_F \rangle$ gilt $f \in FP$ und $\langle F \rangle \in SAT \Leftrightarrow f(\langle F \rangle) \in BSA$

$$\Rightarrow \text{SAT} \leq \text{3SAT}$$
