



1.4 Kontextsensitive und Typ 0-Sprachen



Kuroda Normalform

Eine Typ 1 Grammatik $G = (V, \Sigma, P, S)$ in Kuroda Normalform falls

$$P \subseteq V \times (V \cup \Sigma \cup V^2) \cup V^2 \times V^2$$

Satz: $\forall G$ vom Typ 1 : $\varepsilon \notin L(G) \longrightarrow$

$$\exists G' \text{ in Kuroda Normalform mit } L(G) = L(G')$$

Beweis: nicht hier

Idee: Verallgemeinerung der Chomsky-Normalform



Turing Maschinen

Sind endliche Automaten das letzte Wort?

- + Ein Digitalrechner mit endlich viel Speicher ist ein endlicher Automat!
- Viel Speicher \rightsquigarrow astronomisch viele Zustände, sehr komplexe Automaten
- Wir wollen einfaches Maschinenmodell für Automaten die z.B. $a^n b^n c^n$ akzeptieren.

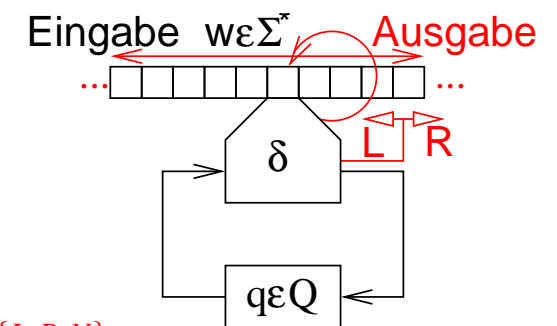


Deterministische Einband-Turingmaschinen

(DTM)

$T = (Q, \Sigma, \Gamma, \delta, s, F)$:

- Q , Zustände
- Σ , Eingabealphabet
- Γ Bandalphabet,
 $\sqcup \notin \Sigma$: Leersymbol,
 $\Sigma \cup \{\sqcup\} \subseteq \Gamma$
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$,
Übergangsfunktion;
- $s \in Q$, Startzustand
- $F \subseteq Q$, Endzustände

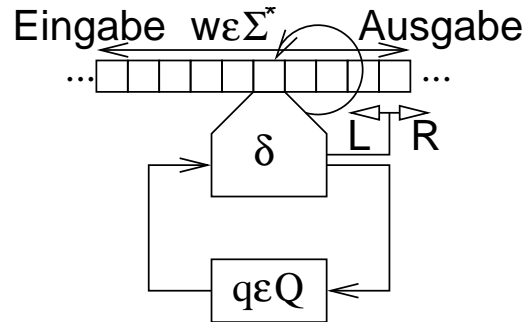




Nichtdeterministische Turingmaschinen (NTM)

$T = (Q, \Sigma, \Gamma, \delta, s, F)$:

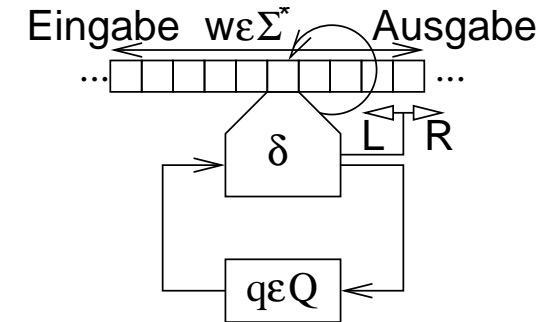
- Q , Zustände
- Σ , Eingabealphabet
- Γ Bandalphabet,
 - $\sqcup \notin \Sigma$: Leersymbol,
 - $\Sigma \cup \{\sqcup\} \subseteq \Gamma$
- $\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L, R, N\}}$,
Übergangsfunktion;
- $s \in Q$, Startzustand
- $F \subseteq Q$, Endzustände



Nichtdeterministische Turingmaschinen (NTM)

$T = (Q, \Sigma, \Gamma, \delta, s, F)$:

- Q , Zustände
- Σ , Eingabealphabet
- Γ Bandalphabet,
 - $\sqcup \notin \Sigma$: Leersymbol,
 - $\Sigma \cup \{\sqcup\} \subseteq \Gamma$
- $\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R, N\}$,
Übergangsrelation;
- $s \in Q$, Startzustand
- $F \subseteq Q$, Endzustände



Warum Turingmaschinen?

- Historisch der erste Ansatz
[Turing 1936]
- Ursprüngliche Motivation:
Reduktion der Arbeitsweise
eines menschlichen Mathematikers
beim Rechnen auf das Wesentliche
- Minimalistische Erweiterung
eines endlichen Automaten
- Churchs These:
Alle „hinreichend mächtigen“
Maschinenmodelle sind äquivalent



Ursprüngliche Motivation

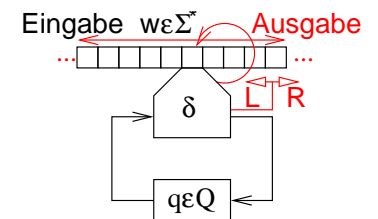
Analog \rightsquigarrow diskrete Positionen, endliches Alphabet: begrenzte Genauigkeit

Stift auf Papier \rightsquigarrow Schreib-Lesekopf

Blatt Papier (2D) \rightsquigarrow Band (1D):

vertikale Bewegungen \rightsquigarrow
größere horizontale Bewegungen,
ggf. Zeilenendmarkierungen

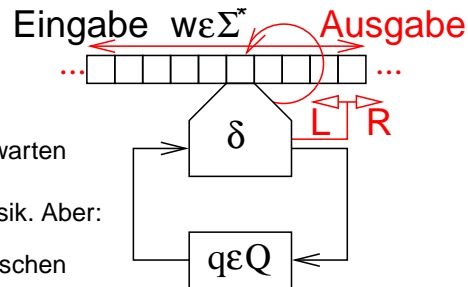
Gehirn befolgt Rechenvorschriften \rightsquigarrow Endlicher Automat



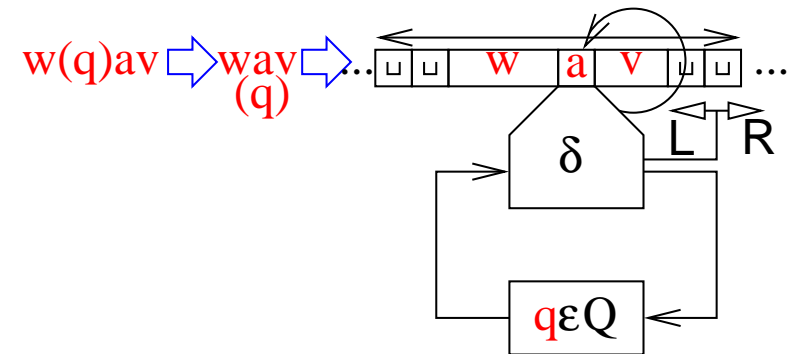


Potentiell unendlicher Speicher?

- + Einzige Alternative zu monströsen endlichen Automaten
- + Blanks an Anfang und Ende nicht abspeichern
- + Wenn Platz ausgeht
mehr „Band“ kaufen
- + Wenn Band ausverkauft
auf nächste **Technologiestufe** warten
- Irgendwann kriegt uns die Physik. Aber:
 - + Dann ist unsere Sonne erloschen
 - + Bis dahin ist das
eine nützliche **Abstraktion**



Konfiguration einer TM



$$w, v \in \Gamma^*, a \in \Gamma, q \in Q$$

Schöning läßt Klammern weg.



Funktionsweise DTM

$$\begin{aligned} \delta(q,b) &= (q',b',N) \\ wa(q)bcv &\vdash wa(q')b'cv \\ \delta(q,b) &= (q',b',L) \\ wa(q)bcv &\vdash w(q')ab'cv \\ \delta(q,b) &= (q',b',R) \\ wa(q)bcv &\vdash wab'(q')cv \end{aligned}$$

Übergangsfunktion δ hat drei Ausgaben:

- Neuer **Zustand** wie bei EA
- Neues **Bandsymbol** — überschreibt altes Symbol an Kopfposition
- Bewegungs**richtung** des Kopfes



Funktionsweise NTM

$$\begin{aligned} (q',b',N) &\in \delta(q,b) \\ wa(q)bcv &\vdash wa(q')b'cv \\ (q',b',L) &\in \delta(q,b) \\ wa(q)bcv &\vdash w(q')ab'cv \\ (q',b',R) &\in \delta(q,b) \\ wa(q)bcv &\vdash wab'(q')cv \end{aligned}$$

mögliche Übergänge zwischen **Konfigurationen**



Wann hält ein DTM?

T hält in Konfiguration $w(q)av$ gdw

$$\delta(q, a) = (q, a, N).$$

Konvention:

$$\forall q \in F : \forall a \in \Gamma : \delta(q, a) = (q, a, N)$$

(nicht im Schöning ?)



Wann hält ein NTM?

T hält in Konfiguration $w(q)av$ gdw

$$\delta(q, a) = \{\}$$

Konvention:

$$\forall q \in F : \forall a \in \Gamma : \delta(q, a) = \{\}$$



Graphinterpretation

$T = (Q, \Sigma, \Gamma, \delta, s, F)$ definiert

unendlichen Multigraphen

Knoten: Konfigurationen von T .

Kanten: von δ zugelassene Konfigurationsübergänge.

$$w \in L(A) \Leftrightarrow \exists \text{ Pfad } P = (s)w \rightarrow \dots \rightarrow u(f)v : f \in F$$

Unterschied DTM versus NTM:

δ bestimmt Konfigurationsübergänge versus

δ läßt Konfigurationsübergänge zu.



Turingmaschinen als Akzeptor

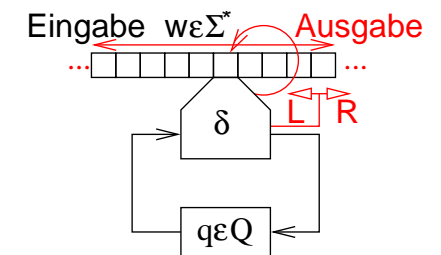
$T = (Q, \Sigma, \Gamma, \delta, s, F)$.

$L(T)$?

T akzeptiert $w \Leftrightarrow$

$$\exists \alpha, \beta \in \Gamma^*, f \in F : (s)w \vdash^* \alpha f \beta$$

$$L(T) := \{w \in \Sigma^* : T \text{ akzeptiert } w\}.$$





Graphinterpretation

$T = (Q, \Sigma, \Gamma, \delta, s, F)$.

$L(T)?$

Definition:

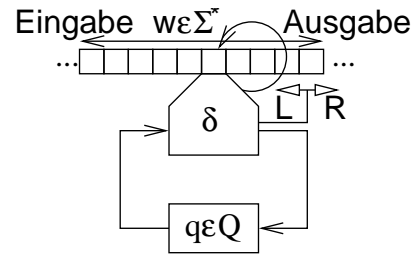
T akzeptiert $w \in \Sigma^*$ gdw

\exists Folge von (durch δ zugelassenen)

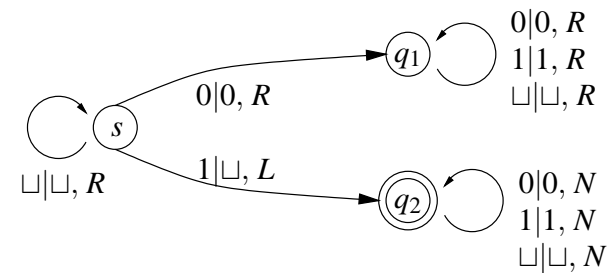
Konfigurationsübergangen

$(s)w \rightarrow \dots \rightarrow x(f)y$ mit $f \in F$.

$L(T) := \{w \in \Sigma^* : T \text{ akzeptiert } w\}$.



Beispiel: Akzeptor für $L = 1(0, 1)^*$



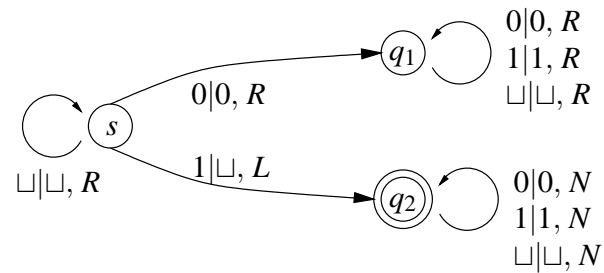
Grafische Notation:

Erweiterung von EA. Erweiterte Kantenbeschriftung:

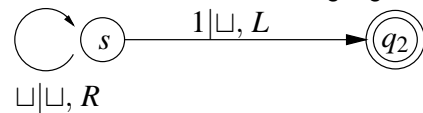
Eingabezeichen|Ausgabezeichen,Bewegungsrichtung



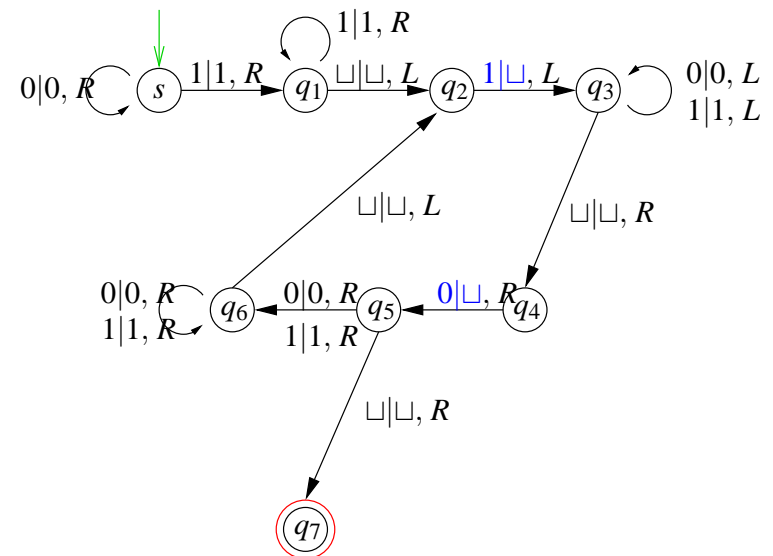
Vervollständigung



Konvention: Fehlender Übergang \rightsquigarrow die TM hält.

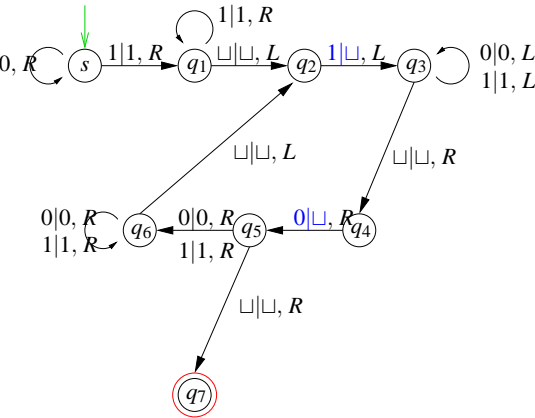


Beispiel: Akzeptor für $\{0^n 1^n : n \geq 1\}$

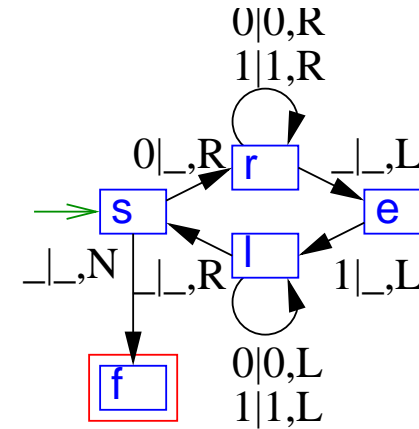




(s)000111	(q ₄)00011	0(q ₂)1	(q ₇)
0(s)00111	⊔(q ₅)0011	(q ₃)0⊔	
00(s)0111	0(q ₆)011	(q ₃)⊔0	
000(s)111	00(q ₆)11	(q ₄)0	
0001(q ₁)11	001(q ₆)1	⊔(q ₅)	
00011(q ₁)1	0011(q ₆)		
000111(q ₁)	001(q ₂)1	0 0, R	
00011(q ₂)1	00(q ₃)1⊔		
0001(q ₃)1⊔	0(q ₃)01		
0001(q ₃)1	(q ₃)001		
000(q ₃)11	(q ₃)⊔001		
00(q ₃)011	(q ₄)001		
0(q ₃)0011	⊔(q ₅)01		
(q ₃)00011	0(q ₆)1		
(q ₃)⊔00011	01(q ₆)		



Beispiel: $\{0^n 1^n : n \geq 0\}$.



Beispiel: $\{0^n 1^n : n \geq 0\}$.

Sei $k \geq 1, w \in \{0, 1\}^*$

ε : $(s) \vdash (f)$.

0: $(s)0 \vdash (r) \vdash (e)$ hält.

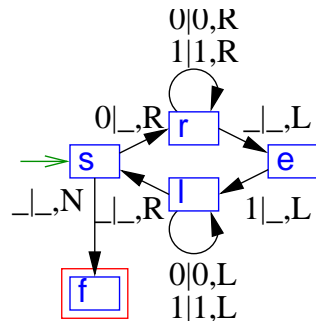
1w: $(s)1w$ hält.

0w0: $(s)0w0 \vdash (r)w0 \vdash w0(r) \vdash w(e)0$ hält.

0w1: $(s)0w1 \vdash (r)w1 \vdash w1(r) \vdash w(e)1 \vdash (\ell) \sqcup w \vdash (s)w$

$0^n 1^n$: $(s)0^n 1^n \vdash^* (s)0^{n-1} 1^{n-1} \vdash^* \dots \vdash^* (s) \vdash (f)$

$0u1, u \notin \{0^n 1^n : n \geq 0\}$: $(s)0u1 \vdash^* (s)u$. Hält nicht in f . (Induktion)



Varianten von Turingmaschinen

k Köpfe: $\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, N\}^k$

k Bänder: i.allg. ein Kopf pro Band

d -dimensionale Bänder: z.B. $d = 2$,

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, U, D, N\}$

probabilistisch: zusätzliches unidirektionales Leseband mit Zufallsbits

**LBA:****Linear Beschränkte Nichtdet. Turingmaschinen**

NTM $T = (Q, \Sigma, \Gamma, \delta, s, F)$ ist **linear beschränkt**, wenn

$$\forall a = a_1 \cdots a_n \in \Sigma^+ : (s)a \vdash^* \alpha(q)\beta \longrightarrow |\alpha\beta| \leq n$$



Satz: \forall Typ 1 Sprachen $L : \exists$ LBA $T : L(T) = L$

Beweisskizze: Sei $G = (V, \Sigma, P, S)$ Typ 1 Grammatik mit $L(G) = L$.

Betrachte die NTM $T = (Q, \Sigma, (\Sigma \cup V), \delta, s, F)$:

Procedure inL(z) // Anfangskonfiguration (s) z

invariant "tape content" $\stackrel{*}{\Rightarrow} z$

invariant $|\text{tape}| \leq |z|$

while tape $\neq S$ **do**

if $\exists w \rightarrow \alpha \in P : \text{tape} = x\alpha y$ **then** // 2× nondet. Choice!

 tape := xwy // contracting rule!

else reject z

 accept z

akzeptierende Berechnung $\xrightarrow{\text{Inv.}} \exists$ Ableitung.

$S \stackrel{*}{\Rightarrow} z \longrightarrow \exists$ korrespondierende akz. Berechnung.

**Unterprogramm: Passende linke Seite suchen**

- OBdA: G ist in Kuroda-Normalform \longrightarrow $|\text{rechte Seite}| \leq 2$
- laufe zum linken Rand
- laufe nach rechts über das Band:
 - Zustand speichert Bandsymbol L links von Lesekopf
 - δ kann abhängig von L , aktuellem Bandsymbol und Kenntnis von P (endlich gross !) feststellen ob es eine passende Produktion gibt
 - Ja? \longrightarrow Ersetzung vornehmen
 - Nein? Weiter

**Unterprogramm: Ersetzung für $AB \rightarrow CD$**

- Einen Schritt nach links
- Schreibe C
- Einen Schritt nach rechts
- Schreibe D
- Zurück zur Hauptschleife

**Unterprogramm: Ersetzung für $AB \rightarrow C$** Schreibe C ; einen Schritt nach linksschreibe \sqcup

laufe zum linken Rand

speichere erstes Bandsymbol im Zustand

repeat

tausche gespeichertes und aktuelles Bandsymbol; einen Schritt nach rechts

until \sqcup überschrieben**Phase 1: generiere Wort aus Σ^*** $T = (Q, \Sigma, \Gamma, \delta, s, F)$ LBA mit $L(T) = L$. $G = (V = \{S\} \cup (\Gamma \times \Sigma) \cup (Q \times \Gamma \times \Sigma), \Sigma, P, S)$. $\{S \rightarrow S(a, a) : a \in \Sigma\} \subseteq P$ $\{S \rightarrow (s, a, a) : a \in \Sigma\} \subseteq P$

Ende Phase 1.

(und Spezialbehandlung falls $\varepsilon \in L$)Beispiel: $S \Rightarrow S(c, c) \Rightarrow S(b, b)(c, c) \Rightarrow (s, a, a)(b, b)(c, c)$ entspricht Anfangskonfiguration $(s)abc$ **Satz:** $\forall L : \exists \text{LBA } T : L(T) = L \rightarrow L$ ist Typ 1 Sprache.**Beweisskizze:** Sei $T = (Q, \Sigma, \Gamma, \delta, s, F)$ LBA mit $L(T) = L$.

Betrachte die Typ 1 Grammatik

 $G = (V = \{S\} \cup (\Gamma \times \Sigma) \cup (Q \times \Gamma \times \Sigma), \Sigma, P, S)$.Idee: TM-Konfiguration $\alpha(q)a\beta \rightsquigarrow$ Satzform $\alpha(q, a)\beta$ plus Extrainfo für ursprüngliche Eingabe.

3 Phasen der Ableitung:

1. **generiere** Wort aus Σ^* .
2. **simuliere** Berechnung der TM.
3. Nach Akzeption **regeneriere** Eingabewort

**Phase 2: **simuliere** Berechnung der TM** $T = (Q, \Sigma, \Gamma, \delta, s, F)$ LBA mit $L(T) = L$. $G = (V = \{S\} \cup (\Gamma \times \Sigma) \cup (Q \times \Gamma \times \Sigma), \Sigma, P, S)$. $P := P \cup \{(q, a, c) \rightarrow (q', a', c) : (q', a', N) \in \delta(q, a), c \in \Sigma\}$ $\cup \{(b, c')(q, a, c) \rightarrow (q', b, c')(a', c) : (q', a', L) \in \delta(q, a), c \in \Sigma\}$ $\cup \{(q, a, c)b \rightarrow a'(q', b, c) : (q', a', R) \in \delta(q, a), c \in \Sigma\}$ Beispiel: $(s, a, a)(b, b)(c, c) \xRightarrow{*} (x, a)(f, y, b)(z, c)$ entspricht Konfigurationsfolge $(s)abc \vdash \dots \vdash x(f)yz$



Phase 3: regeneriere Eingabewort

$T = (Q, \Sigma, \Gamma, \delta, s, F)$ LBA mit $L(T) = L$.

$G = (V = \{S\} \cup (\Gamma \times \Sigma) \cup (Q \times \Gamma \times \Sigma), \Sigma, P, S)$.

$\{(f, a, c) \rightarrow c : f \in F, a \in \Gamma, c \in \Sigma\} \subseteq P$

$\{(a, b) \rightarrow b : a \in \Gamma \wedge b \in \Sigma\} \subseteq P$

Akzeption.

Beispiel: $(x, a)(f, y, b)(z, c) \Rightarrow (x, a)b(z, c) \Rightarrow ab(z, c) \Rightarrow abc$



Abschlusseigenschaften Typ 1

Abgeschlossen unter

Vereinigung

Produkt

Stern

Schnitt

Komplement

U

.

*

∩

⊖



Abschluss Vereinigung Typ 0/1

$L(A_1) \cup L(A_2)$ hat Typ 0/1 ?

Zwei Unterprogramme U_1 für $L(A_1)$ bzw. U_2 für $L(A_2)$.

Neues Programm für $U_1 \vee U_2$.

Kein zusätzlicher Platzverbrauch.

Übung: Abschluss Schnitt Typ 0/1



Abschluss Produkt Typ 0/1

$L(A_1) \cdot L(A_2)$ hat Typ 0/1 ?

Zwei Unterprogramme U_1 für $L(A_1)$ bzw. U_2 für $L(A_2)$.

Neues Programm für $w \in L(A_1) \cdot L(A_2)$?:

Spalte $w = w_1 w_2$ (nichtdeterministisch).

return $w_1 \in L(A_1) \wedge w_2 \in L(A_2)$

Geht ohne Bänderweiterung mittels zusätzlicher Bandmarkierungen.

Übung: Abschluss Kleensche Hülle (*) Typ 0/1



Typ	Beschreibungsmittel
3	reichtlineare oder linkslineare Grammatik DFA, $\bar{\epsilon}$ NFA, ϵ NFA regulärer Ausdruck
Det. KF	LR(k) Grammatik DKellerA mit Endz.
2	kontextfreie Grammatik (1Zustands)NKellerA
1	kontextsensitive Grammatik LBA
0	Typ 0 Grammatik Turingmaschine



Typ	Nichtdet.	Deterministisch	äquivalent?
3	NFA	DFA	Ja
2	NKellerA	DKellerA	Nein
1	LBA	DLBA	???
0	NTM	DTM	Ja (bez. Berech.)



Abschlusseigenschaften

Typ	\cap	\cup	$\bar{\cdot}$	\cdot	*
3	ja	ja	ja	ja	ja
Det. KF	nein	nein	ja	nein	nein
2	nein	ja	nein	ja	ja
1	ja	ja	ja	ja	ja
0	ja	ja	nein	ja	ja



Entscheidbarkeitsprobleme

Typ	Wort-	Leerheits-	Äquivalenz	Schnitt leer
3	ja	ja	ja	ja
Det. KF	ja	ja	ja [97]	nein
2	ja	ja	nein	nein
1	ja	nein	nein	nein
0	nein	nein	nein	nein



Komplexität des Wortproblems

Typ	Beschreibungsmittel
3	$\mathcal{O}(n)$
Det. KF	$\mathcal{O}(n)$
2	$\mathcal{O}(n^3)$
1	$ \Sigma ^{\mathcal{O}(n)}$, "NP-hart" \rightsquigarrow Komplexitätstheorie
0	"semientscheidbar" \rightsquigarrow Berechenbarkeit



Chomsky-Hierarchie: Eine Kritik

- 2: (Nur?) hier sind Grammatiken "genau richtig"
- 3: "zufällig" machen lineare Produktionen das gleiche wie ein endlicher Automat
- 0,1: kontextsensitive Regeln sind zu low level und zu ähnlich zu TM um interessantes Modellierungswerkzeug zu sein
- 1: Einer von vielen entscheidbaren Spezialfällen?
Warum ausgerechnet lineare Platzbeschränkung?
Es gibt nützlichere Verallgemeinerungen von KFGs.