# 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 : $\varepsilon \notin L(G) \longrightarrow$

$\exists G'$ 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!

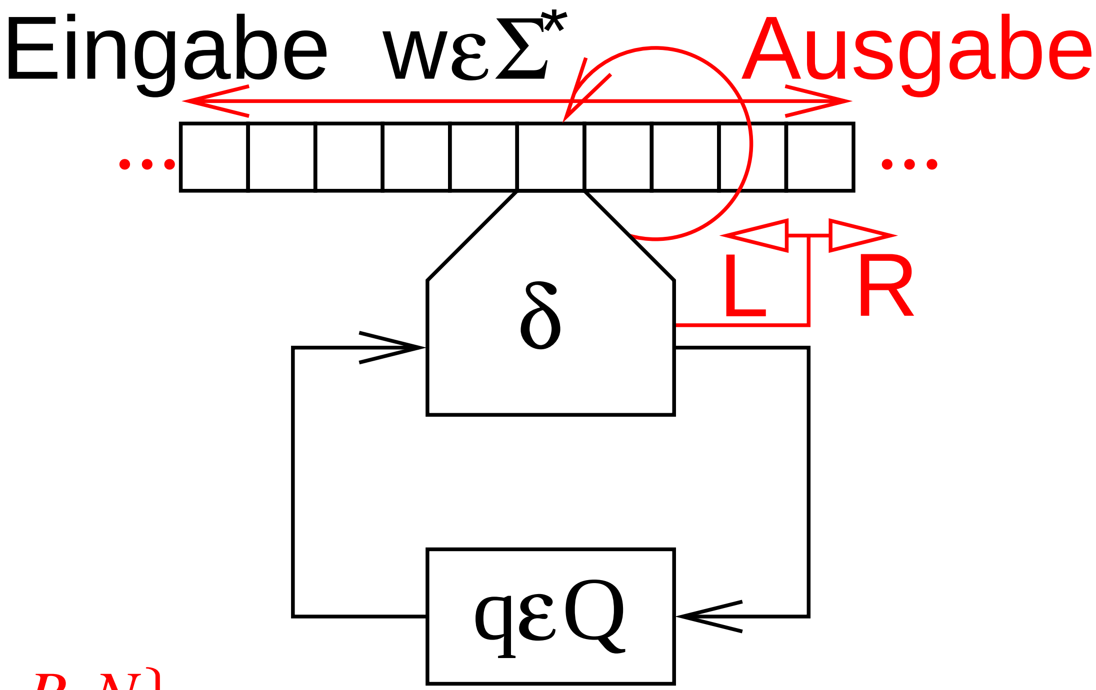
$-$: Viel Speicher $\rightsquigarrow$ astronomisch viele Zustände, sehr komplexe Automaten

$-$: Wir wollen einfaches Maschinenmodell für Automaten die z.B. $a^n b^n c^n$ akzeptieren.

# Deterministische Einband-Turingmaschinen (DTM)

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

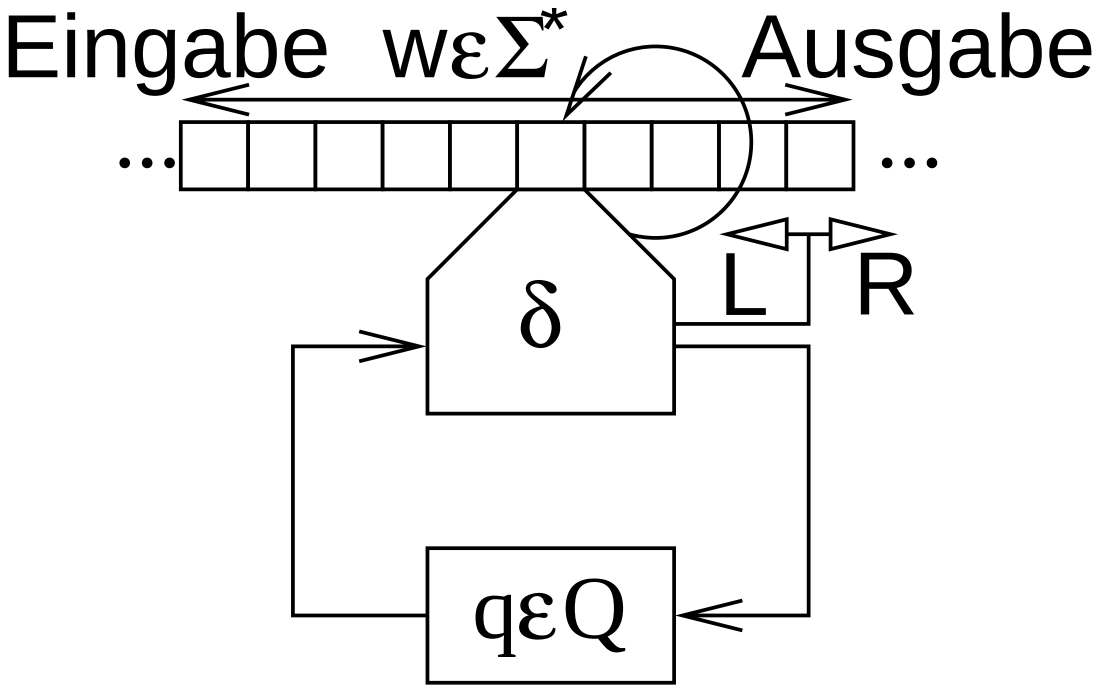
- $Q$, Zustände
- $\Sigma$, Eingabealphabet
- $\Gamma$ Bandalphabet,
  $\sqcup \notin \Sigma$: Leersymbol,
  $\Sigma \cup \{\sqcup\} \subseteq \Gamma$
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$,
  Übergangsfunktion;
- $s \in Q$, Startzustand
- $F \subseteq Q$, Endzustände



# Nichtdeterministische Turingmaschinen (NTM)

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

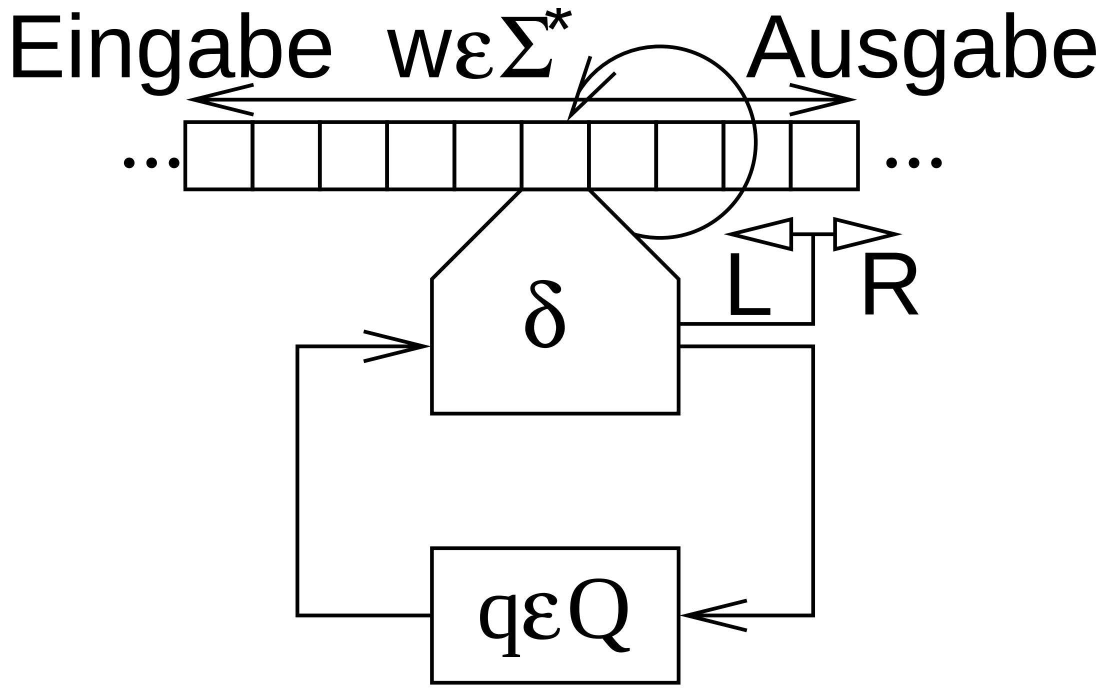
- $Q$, Zustände
- $\Sigma$, Eingabealphabet
- $\Gamma$ Bandalphabet,
  $\sqcup \notin \Sigma$: Leersymbol,
  $\Sigma \cup \{\sqcup\} \subseteq \Gamma$
- $\delta : Q \times \Gamma \to 2^{Q \times \Gamma \times \{L,R,N\}}$,
  Übergangsfunktion;
- $s \in Q$, Startzustand
- $F \subseteq Q$, Endzustände



# Nichtdeterministische Turingmaschinen (NTM)

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

- $Q$, Zustände
- $\Sigma$, Eingabealphabet
- $\Gamma$ 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
- Curchsche These:
  Alle „hinreichend mächtigen“
  Maschinenmodelle sind äquivalent

# Ursprüngliche Motivation

Analog $\rightsquigarrow$ diskrete Positionen, endliches Alphabet: begrenzte Genauigkeit

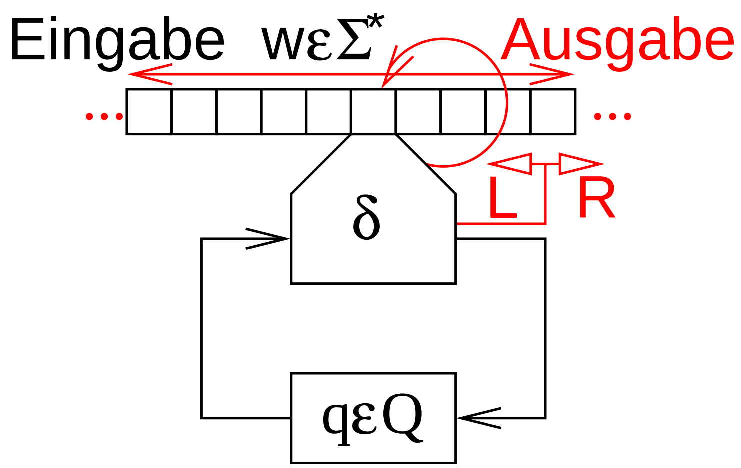
Stift auf Papier $\rightsquigarrow$ Schreib-Lesekopf

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

vertikale Bewegungen $\rightsquigarrow$

größere horizontale Bewegungen,

ggf. Zeilenendemarkierungen




Gehirn befolgt Rechenvorschriften $\rightsquigarrow$ Endlicher Automat

# Potentiell unendlicher Speicher?

+: Einzige Alternative zu monströsen endlichen Automaten

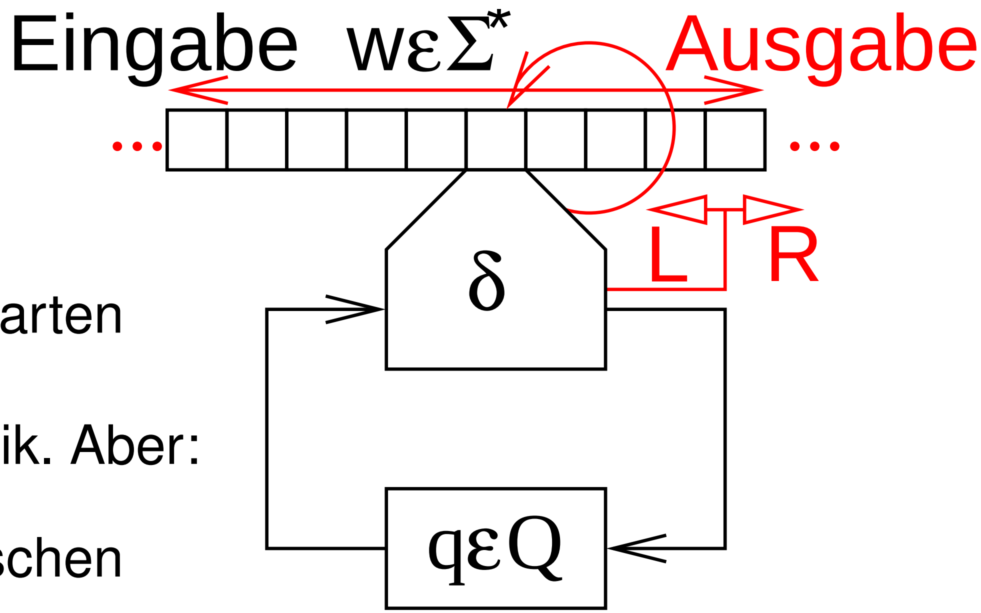
+: Blanks an Anfang und Ende nicht abspeichern

+: Wenn Platz ausgeht
mehr „Band“ kaufen

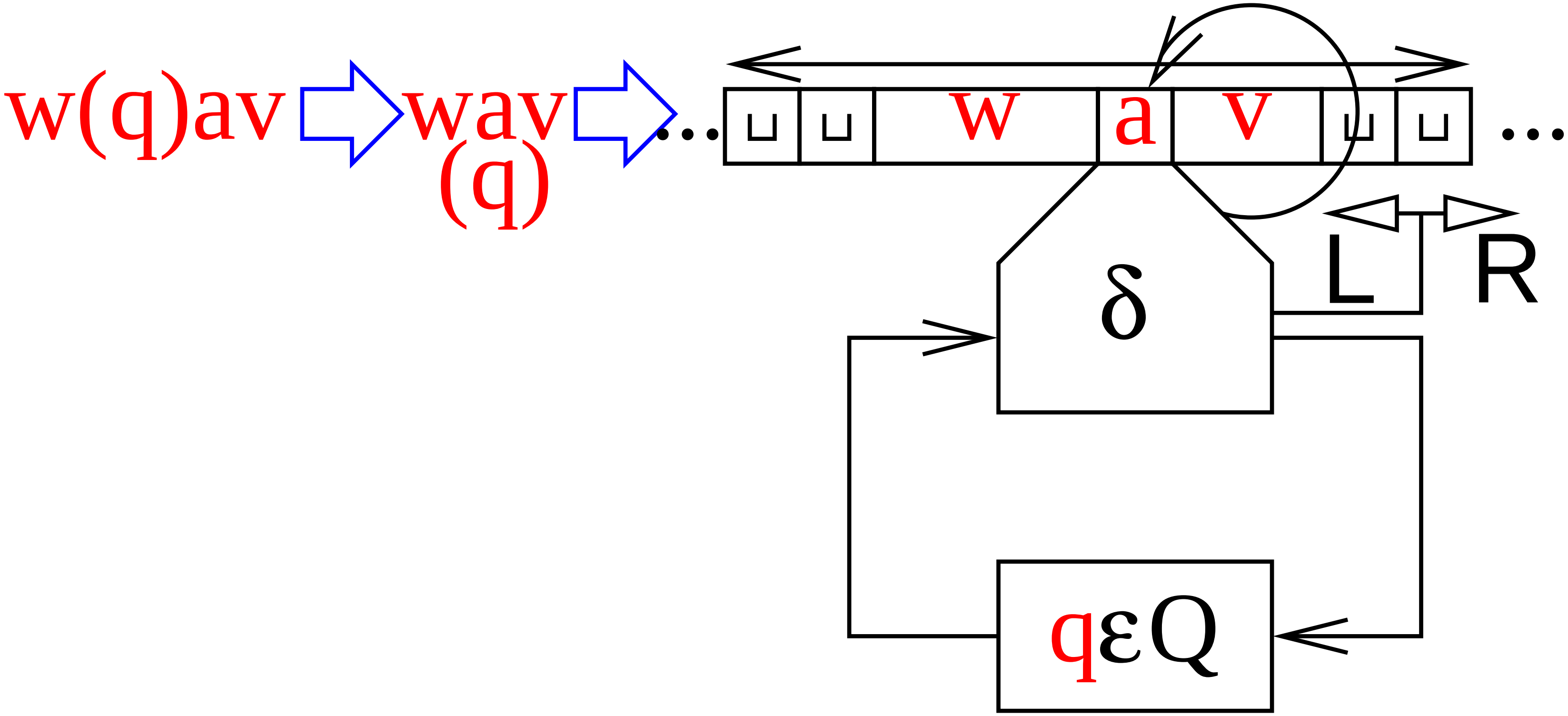
+: Wenn Band ausverkauft
auf nächste Technologiestufe warten

−: 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

$$wa(q)bcv \overset{\delta(q,b)=(q',b',N)}{\vdash} wa(q')b'cv$$

$$wa(q)bcv \overset{\delta(q,b)=(q',b',L)}{\vdash} w(q')ab'cv$$

$$wa(q)bcv \overset{\delta(q,b)=(q',b',R)}{\vdash} wab'(q')cv$$

Übergangsfunktion $\delta$ hat drei Ausgaben:

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

# Funktionsweise NTM

$$wa(q)bcv \overset{(q',b',N)\in\delta(q,b)}{\vdash} wa(q')b'cv$$

$$wa(q)bcv \overset{(q',b',L)\in\delta(q,b)}{\vdash} w(q')ab'cv$$

$$wa(q)bcv \overset{(q',b',R)\in\delta(q,b)}{\vdash} wab'(q')cv$$

mögliche Übergänge zwischen Konfigurationen

# Wann hält eine 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 eine 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 $\delta$ zugelassene Konfigurationsübergänge.

$w \in L(A) \Leftrightarrow \exists$ Pfad $P = (s)w \to \cdots \to u(f)v : f \in F$

Unterschied DTM versus NTM:

$\delta$ bestimmt Konfigurationsübergänge versus

$\delta$ lässt Konfigurationsübergänge zu.

# Turingmaschinen als Akzeptor

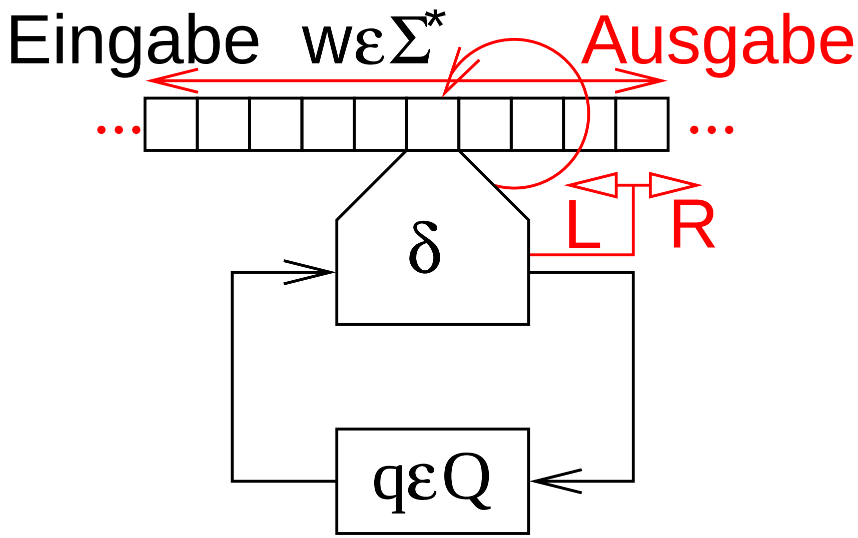
$T = (Q, \Sigma, \Gamma, \delta, s, F)$.

$L(T)$?

$T$ 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:**

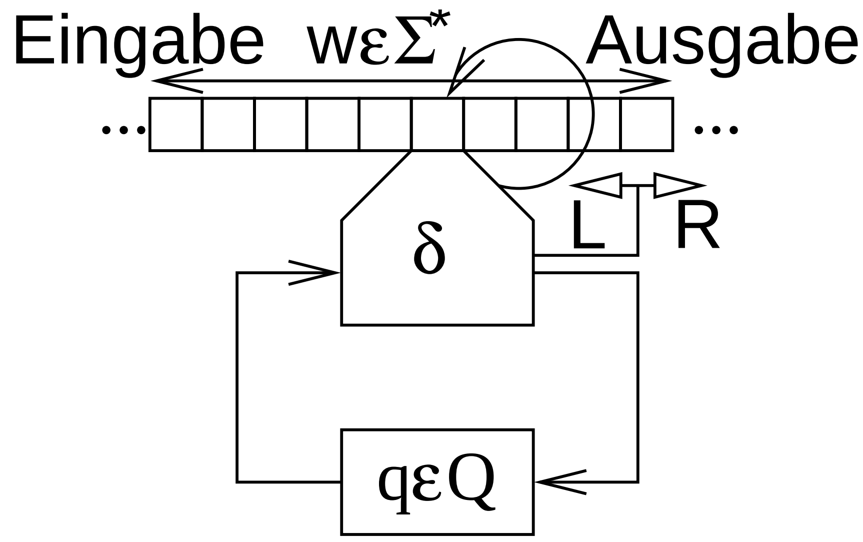
$T$ akzeptiert $w \in \Sigma^*$ gdw

$\exists$ Folge von (durch $\delta$ zugelassenen)

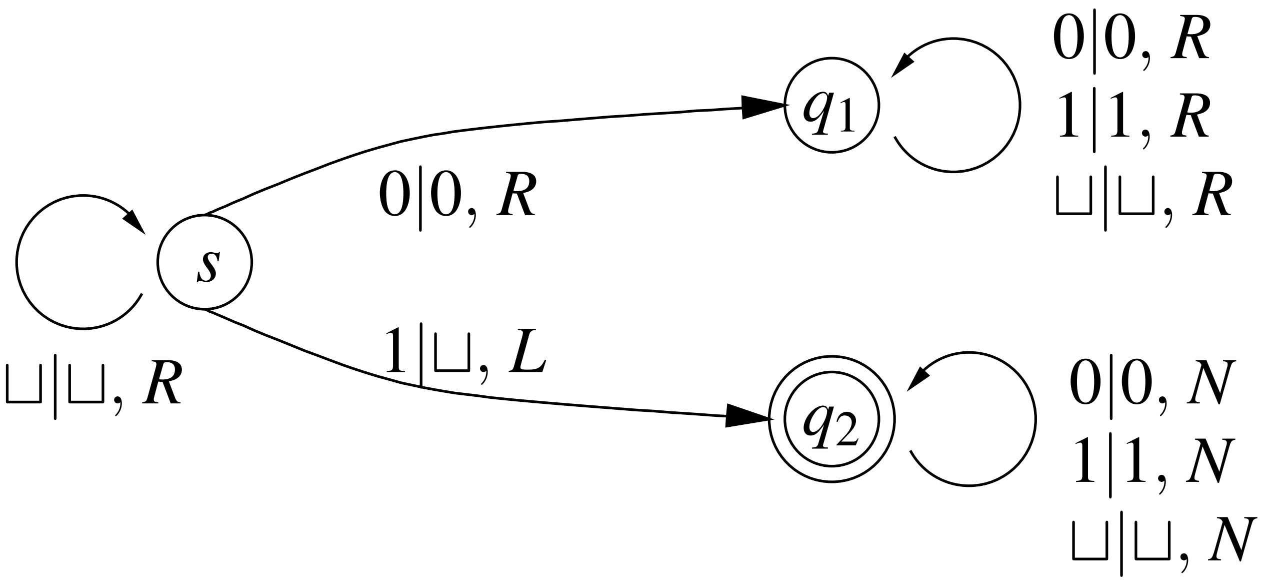
Konfigurationsübergangen

$(s)w \to \cdots \to x(f)y$ mit $f \in F$.

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



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

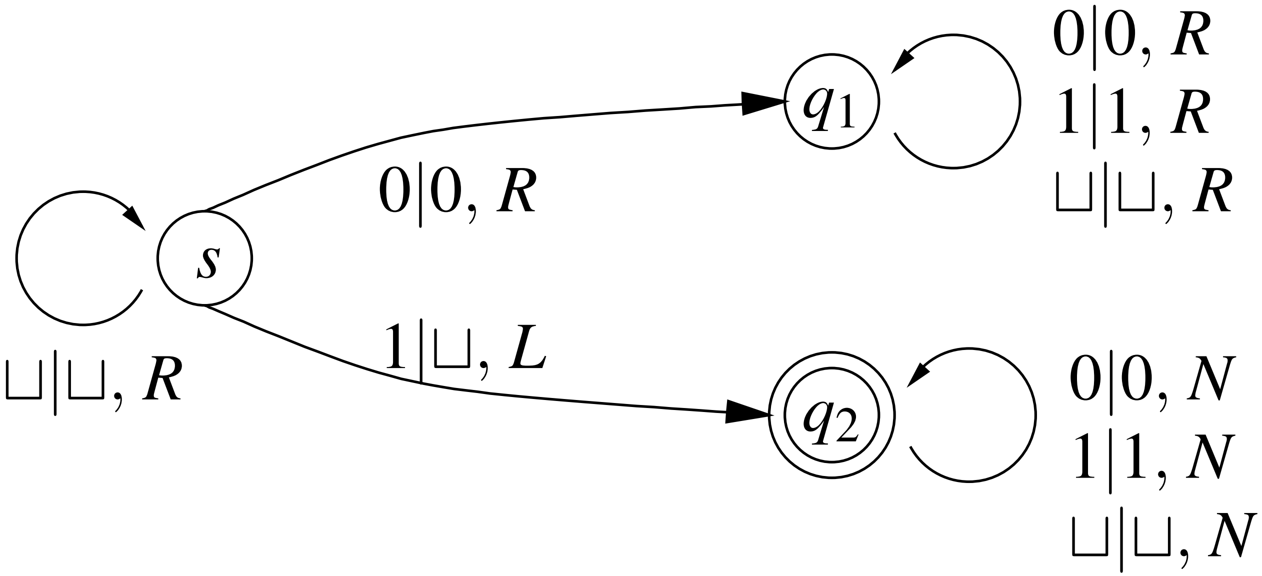



## Grafische Notation:

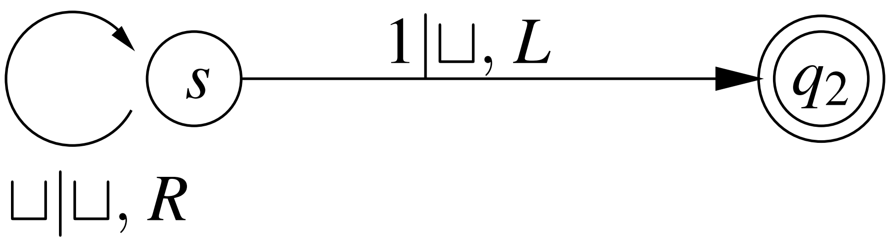
Erweiterung von EA. Erweiterte Kantenbeschriftung:

Eingabezeichen|Ausgabezeichen,Bewegungsrichtung

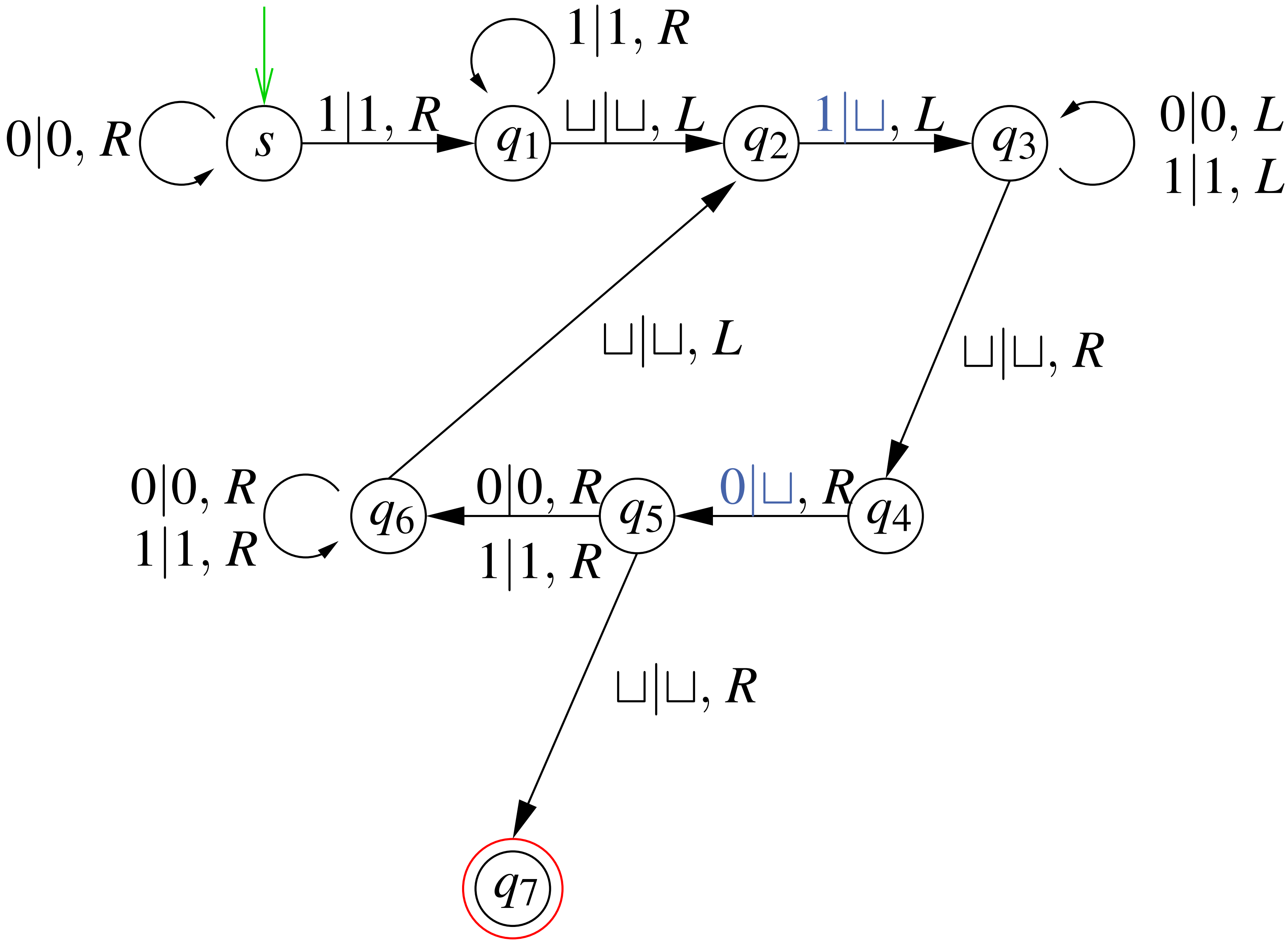
# Vervollständigung




Konvention: Fehlender Übergang $\rightsquigarrow$ die TM hält.



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

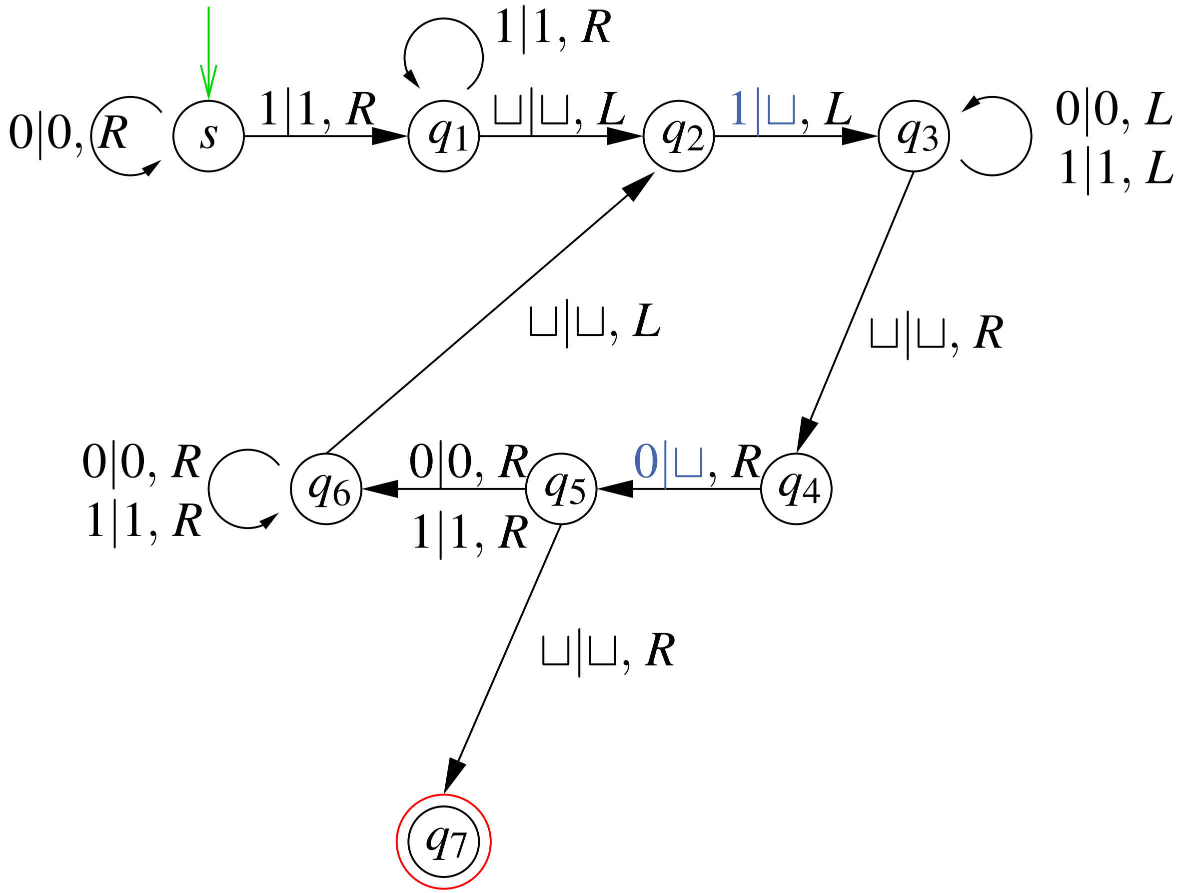


$(s)000111$
$0(s)00111$
$00(s)0111$
$000(s)111$
$0001(q_1)11$
$00011(q_1)1$
$000111(q_1)$
$00011(q_2)1$
$0001(q_3)1\sqcup$
$0001(q_3)1$
$000(q_3)11$
$00(q_3)011$
$0(q_3)0011$
$(q_3)00011$
$(q_3)\sqcup 00011$

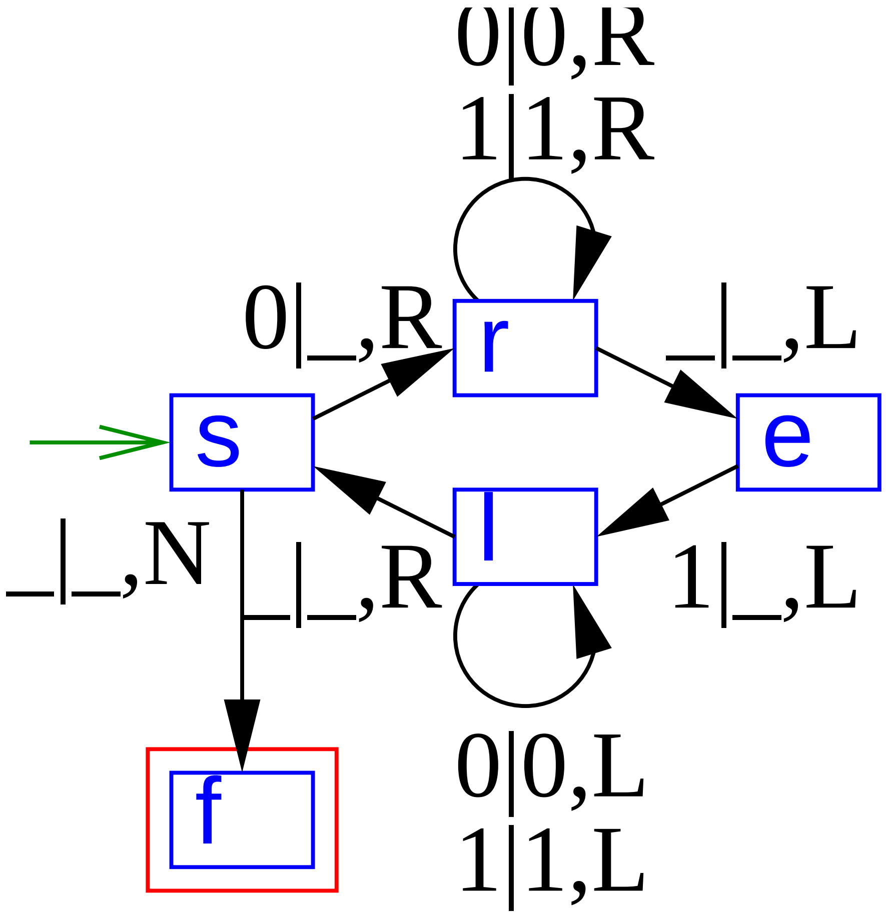
$(q_4)00011$
$\sqcup(q_5)0011$
$0(q_6)011$
$00(q_6)11$
$001(q_6)1$
$0011(q_6)$
$001(q_2)1$
$00(q_3)1\sqcup$
$0(q_3)01$
$(q_3)001$
$(q_3)\sqcup 001$
$(q_4)001$
$\sqcup(q_5)01$
$0(q_6)1$
$01(q_6)$

$0(q_2)1$
$(q_3)0\sqcup$
$(q_3)\sqcup 0$
$(q_4)0$
$\sqcup(q_5)$

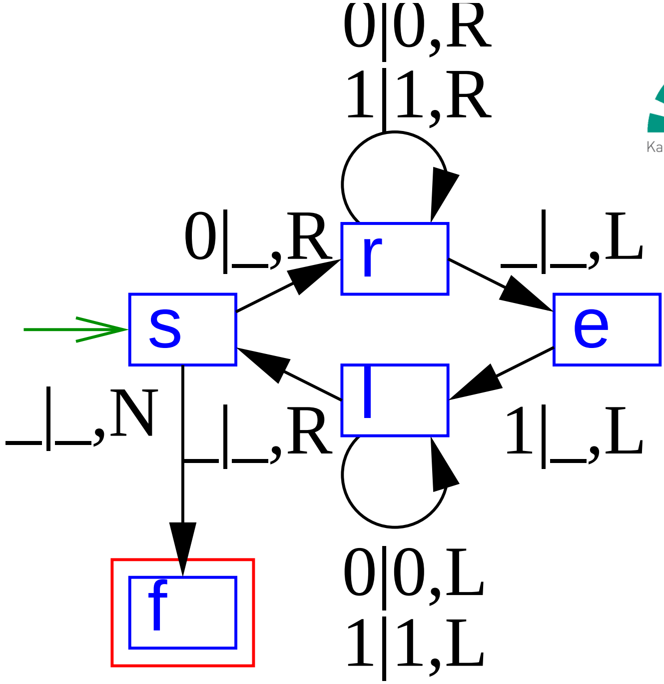
$(q_7)$



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



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




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

$\varepsilon$: $(s) \vdash (f)$.

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

$1w$: $(s)1w$ hält.

$0w0$: $(s)0w0 \vdash (r)w0 \overset{|w|+1}{\vdash} w0(r) \vdash w(e)0$ hält.

$0w1$: $(s)0w1 \vdash (r)w1 \overset{|w|+1}{\vdash} w1(r) \vdash w(e)1 \overset{|w|+1}{\vdash} (\ell) \sqcup w \vdash (s)w$

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

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

# Varianten von Turingmaschinen

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

$k$ Bänder: i.allg. ein Kopf pro Band

$d$-dimensionale Bänder: z.B. $d = 2$,

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

probabilistisch: zusätzliches undirektionales 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"$\overset{*}{\Rightarrow} z$

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

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

    **if** $\exists w \to \alpha \in P$ : tape $= x\alpha y$ **then** // $2\times$nondet. Choice!

      tape:= $xwy$ // contracting rule!

    **else** reject $z$

  accept $z$

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

$S \overset{*}{\Rightarrow} z \longrightarrow \exists$ 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
  - $\delta$ 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 $AB \to CD$

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

# Unterprogramm: Ersetzung für $AB \to C$

Schreibe $C$; einen Schritt nach links

schreibe $\sqcup$

laufe zum linken Rand

speichere erstes Bandsymbol im Zustand

**repeat**

    tausche gespeichertes und aktuelles Bandsymbol; einen Schritt nach rechts

**until** $\sqcup$ überschrieben

**Satz:** $\forall L$ : $\exists$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 $\Sigma^*$.
2. simuliere Berechnung der TM.
3. Nach Akzeption regeneriere Eingabewort

# Phase 1: generiere Wort aus $\Sigma^*$

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

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

$\{S \to S(a,a) : a \in \Sigma\} \subseteq P$

$\{S \to (s,a,a) : a \in \Sigma\} \subseteq P$

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 $(s)abc$

# Phase 2: simuliere Berechnung der TM

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

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

$$P := P\cup\{(q,a,c)\to(q',a',c) \quad : (q',a',N)\in\delta(q,a), c\in\Sigma\}$$

$$\cup\{(b,c')(q,a,c)\to(q',b,c')(a',c) \quad : (q',a',L)\in\delta(q,a), c\in\Sigma\}$$

$$\cup\{(q,a,c)b\to a'(q',b,c) \quad : (q',a',R)\in\delta(q,a), c\in\Sigma\}$$

Beispiel: $(s,a,a)(b,b)(c,c)\stackrel{*}{\Rightarrow}(x,a)(f,y,b)(z,c)$

entspricht Konfigurationsfolge $(s)abc\vdash\cdots\vdash x(f)yz$

# Phase 3: regeneriere Eingabewort

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

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

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

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

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 $\cup$
- ☐ Produkt $\cdot$
- ☐ Stern $*$
- ☐ Schnitt $\cap$
- ☐ Komplement $\bar{\cdot}$

# 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. $U_2$ $L(A_2)$.

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. $U_2$ $L(A_2)$.

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 Banderweiterung mittels zusätzlicher Bandmarkierungen.

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

# Abschluss Komplement Typ 1

Geg: $L = L(G) \subseteq \Sigma^*$, $G = (V, \Sigma, P, S)$.

Idee: gib LBA $M$ an, die $x \in \Sigma^n$ akzeptiert gdw. $x \notin L$.

$a$:= $\left|\left\{\alpha \in (V \cup \Sigma)^* : |\alpha| \leq n \wedge S \overset{*}{\Rightarrow} \alpha\right\}\right|$ // todo

$c$:= 0

**foreach** $\alpha \in (V \cup \Sigma)^* \setminus \{x\}$ with $|\alpha| \leq n$ **do**

  **if** $S \overset{*}{\Rightarrow} \alpha$ **then** $c{+}{+}$ // nondeterministic

  **else** continue

**return** $c = a$

**Berechne** $\left|\left\{\alpha \in (V \cup \Sigma)^* : |\alpha| \leq n \wedge S \overset{*}{\Rightarrow} \alpha\right\}\right|$

$a := 1$ // $|\{S\}|$

**for** $m := 1$ **to** $\infty$ **do** // Ableitungslänge

$b := 0$ // Zähler für neue oder alte Wörter

**foreach** $w' \in (V \cup \Sigma)^*$ with $|w'| \leq n$ **do**

z:= 0 // Zähler für alte Wörter

**foreach** $w \in (V \cup \Sigma)^*$ with $|w| \leq n$ **do**

**if** $S \overset{\leq m}{\Rightarrow} w$ **then** $z++$ // nondeterministic

**if** $w = w' \vee w \Rightarrow w'$ **then** $b++$ ; break loop

**if** $z \neq a$ **then** fail

**if** $a = b$ **then** break loop

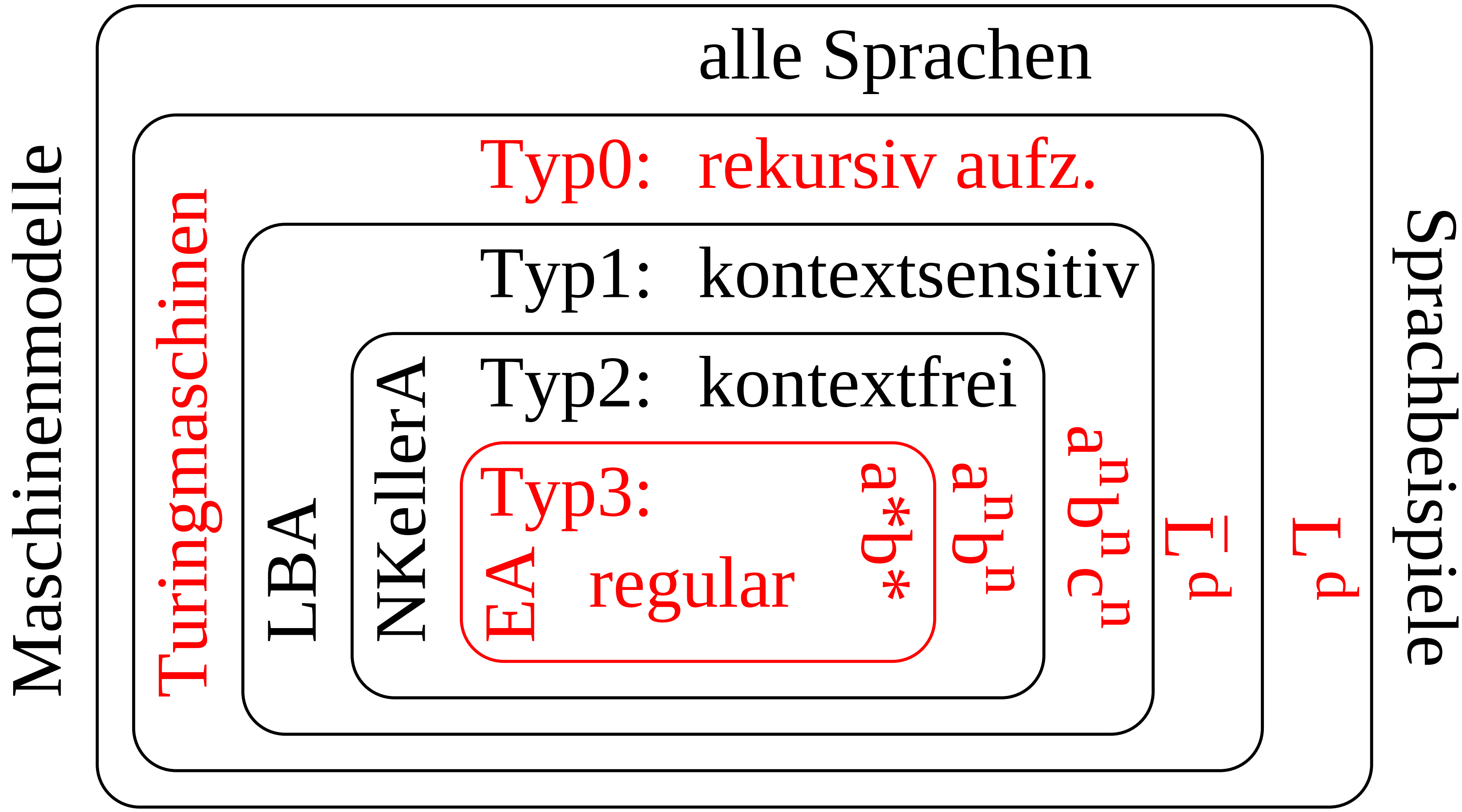
$a := b$

# Typ 0 Sprachen

**Satz:** $L$ durch TM akzeptiert $\Leftrightarrow$ $L$ hat Typ 0

**Beweis:** Analog zum Beweis für Typ 1.

# Überblick Chomsky-Hierarchie



| Typ | Beschreibungsmittel |
|---|---|
| 3 | rechtlineare oder linkslineare Grammatik<br>DFA, $\bar{\varepsilon}$NFA, $\varepsilon$NFA<br>regulärer Ausdruck |
| Det. KF | LR$(k)$ Grammatik<br>DKellerA mit Endz. |
| 2 | kontextfreie Grammatik<br>(1Zustands)NKellerA |
| 1 | kontextsensitive Grammatik<br>LBA |
| 0 | Typ 0 Grammatik<br>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 Wortproblemes

| Typ | Beschreibungsmittel |
|---|---|
| 3 | $\mathcal{O}(n)$ |
| Det. KF | $\mathcal{O}(n)$ |
| 2 | $\mathcal{O}\left(n^3\right)$ |
| 1 | $\lvert\Sigma\rvert^{\mathcal{O}(n)}$, "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.