

3. Die Hierarchie der zustandsänderungsbeschränkten Zellularautomaten

In diesem Kapitel wird eine Hierarchie der Sprachfamilien von Zellularautomaten mit verschiedener Zustandsänderungsbeschränkung aufgestellt und verfeinert.

Anschließend wird ein notwendiges Kriterium für die von zustandsänderungsbeschränkten unidirektionalen Zellularautomaten erkannten Sprachen hergeleitet. Ein entsprechendes Kriterium für bidirektionale Zellularautomaten existiert nur für Realzeitsprachen einer speziellen Struktur.

3.1 Einige grundlegende Hilfssätze

Die folgenden Lemmata folgen unmittelbar aus den Definitionen im vorhergehenden Kapitel. Sie sind bis auf Lemma 3.3 bereits in [Vo81] enthalten und werden zur Vereinfachung der Beweise in den folgenden Abschnitten eingeführt.

Lemma 3.1 Ein $\rho(n)$ -zustandsänderungsbeschränkter Zellularautomat ist $n \cdot \rho(n)$ -zeitbeschränkt. [Vo81]

Beweis: Ein $\rho(n)$ -änderungsbeschränkter Zellularautomat ist $n \cdot \rho(n)$ -sum-änderungsbeschränkt. Da in jedem Schritt wenigstens ein Automat seinen Zustand ändern muß, damit der Zellularautomat nicht in einem Zustand stehenbleibt, ist ein solcher ZA $n \cdot \rho(n)$ -zeitbeschränkt. ◇

Lemma 3.2 Es seien $w, \bar{w} \in V^+$; $v, \mu, x \in \mathbb{N}$ mit $1 \leq v \leq |w| - 1$, $1 \leq \mu \leq |\bar{w}| - 1$. [Vo81]

Sind die beiden x -Verhaltensfolgeausschnitte $\langle c_{\circ}^w(v, v+1) \rangle|_x$ und $\langle c_{\circ}^{\bar{w}}(\mu, \mu+1) \rangle|_x$ eines ZA gleich, so gilt für

$$w^l = w_{1,v} \bar{w}_{\mu+1, |w|} \quad \text{und} \quad \bar{w}^l = \bar{w}_{1,\mu} w_{v+1, |w|} :$$

$$\langle c_{\circ}^{w^l}(1) \rangle|_{x+v} = \langle c_{\circ}^w(1) \rangle|_{x+v},$$

$$\langle c_{\circ}^{\bar{w}^l}(1) \rangle|_{x+\mu} = \langle c_{\circ}^{\bar{w}}(1) \rangle|_{x+\mu}.$$

Beweis: In einem ZA mit H_1 -Raster ist der k -Verhaltensfolgeausschnitt $\langle c_{\circ}^y(v+1) \rangle|_k$ eindeutig bestimmt durch die Anfangskonfiguration $c_{\circ}^y(v+1, |v|)$ und den benachbarten $(k-1)$ -Verhaltensfolgeausschnitt $\langle c_{\circ}^y(v) \rangle|_{k-1}$; der k -Verhaltensfolgeausschnitt $\langle c_{\circ}^y(v) \rangle|_k$ wird eindeutig bestimmt durch die Anfangskonfiguration $c_{\circ}^y(1, v)$ und den benachbarten $(k-1)$ -Verhaltensfolgeausschnitt $\langle c_{\circ}^y(v+1) \rangle|_{k-1}$.

Durch Induktion über k folgt dann $\langle c_{\circ}^{w^l}(v, v+1) \rangle|_x = \langle c_{\circ}^w(v, v+1) \rangle|_x$.

Da w und w^l nach Definition in den ersten v Zeichen übereinstimmen, erhält man die erste Behauptung. Die zweite ergibt sich dann durch einfaches Vertauschen von w und \bar{w} .

◇

Im Fall eines unidirektionalen ZA kann Lemma 3.2 verschärft werden, da nun der Folgezustand eines Automaten unabhängig vom Zustand seines linken Nachbarn ist.

Lemma 3.3 Es seien $w, \bar{w} \in V^+$; $v, \mu, x \in \mathbb{N}$ mit $1 \leq v \leq |w| - 1$, $1 \leq \mu \leq |\bar{w}| - 1$.

Sind die beiden x -Verhaltensfolgeausschnitte $\langle c_{\circ}^w(v+1) \rangle|_x$ und $\langle c_{\circ}^{\bar{w}}(\mu+1) \rangle|_x$ eines unidirektionalen ZA gleich, so gilt für

$$w^l = w_{1,v} \bar{w}_{\mu+1, |\bar{w}|} \quad \text{und} \quad \bar{w}^l = \bar{w}_{1,\mu} w_{v+1, |w|} :$$

$$\langle c_{\circ}^{w^l}(1) \rangle|_{x+v} = \langle c_{\circ}^w(1) \rangle|_{x+v},$$

$$\langle c_{\circ}^{\bar{w}^l}(1) \rangle|_{x+\mu} = \langle c_{\circ}^{\bar{w}}(1) \rangle|_{x+\mu}$$

Beweis: In einem unidirektionalen ZA ist der k -Verhaltensfolgeausschnitt $\langle c_{\circ}^v(v+1) \rangle|_k$ eindeutig bestimmt durch die Anfangskonfiguration $c_{\circ}^v(v+1, |v|)$. Aus der Voraussetzung $\langle c_{\circ}^w(v+1) \rangle|_x = \langle c_{\circ}^{\bar{w}}(\mu+1) \rangle|_x$ und aus $c_{\circ}^{w^l}(v+1, |w^l|) = c_{\circ}^{\bar{w}}(\mu+1, |\bar{w}|)$ folgt demnach $\langle c_{\circ}^w(v+1) \rangle|_x = \langle c_{\circ}^{w^l}(v+1) \rangle|_x$.

Da w und w^l nach Definition in den ersten v Zeichen übereinstimmen, erhält man die erste Behauptung. Die zweite Behauptung folgt wieder durch Vertauschen von w und \bar{w} .

◇

Lemma 3.4 a) In einem ZA gibt es für $\tau, \rho \in \mathbb{N}$ mit $\tau \geq \rho$ nicht mehr als

[Vo82]

$$\binom{\tau}{\rho} |A|^{\rho+1} \leq \tau^{\rho} \cdot |A|^{\rho+1}$$

verschiedene τ -Verhaltensfolgeausschnitte $\langle c_{\circ}(v) \rangle|_{\tau}$ mit höchstens ρ Zustandsänderungen.

b) Die Anzahl der τ -Verhaltensfolgeausschnitte $\langle c_{\circ}(v, v+1) \rangle|_{\tau}$ mit höchstens ρ Zustandsänderungen pro Zelle ist beschränkt durch

$$\left(\binom{\tau}{\rho} |A|^{\rho+1} \right)^2 \leq \tau^{2\rho} \cdot |A|^{2(\rho+1)}.$$

Beweis: Ein τ -Verhaltensfolgeausschnitt kann als Vektor der Länge $\tau+1$ mit Elementen aus A dargestellt werden. An maximal ρ Stellen von 2 bis $\tau+1$ des Vektors können Zustandsänderungen auftreten. Es gibt $|A|$ Möglichkeiten für den Zustand am Anfang des Vektors und $|A|^\rho$ Möglichkeiten für die neuen Zustände nach jedem Zustandswechsel. Insgesamt gibt es also höchstens

$$\binom{\tau}{\rho} |A|^{\rho+1} \leq \tau^\rho |A|^{\rho+1}$$

solche Vektoren. Lemma 3.4 b) folgt dann unmittelbar aus a).

◇

In den Beweisen der nächsten Abschnitte wird auch eine untere Schranke für den Binomialkoeffizienten benötigt, die im folgenden Lemma bewiesen wird.

Lemma 3.5 Es seien $n, k \in \mathbf{N}$ mit $k \leq n^{\frac{1}{2}}$. Dann gilt

$$\binom{n}{k} > n^{\left(\frac{1}{2}k\right)}$$

Beweis: Mit $n! \leq \left(\frac{n}{2}\right)^n$ läßt sich der Binomialkoeffizient nach unten abschätzen durch

$$\binom{n}{k} = \frac{n!}{(n-k)! \cdot k!} \quad \text{nach Definition des Binomialkoeffizienten.}$$

$$= \frac{n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (n-k+1)}{k!}$$

$$> \frac{\left(\frac{n}{2}\right)^k}{\left(\frac{1}{2}k\right)^k} \quad \text{denn alle } k \text{ Faktoren im Zähler sind größer als } \frac{n}{2}.$$

$$= \left(\frac{n}{k}\right)^k \geq \left(n^{\frac{1}{2}}\right)^k \quad \text{wegen } k \leq n^{\frac{1}{2}} \text{ gilt } \frac{n}{k} \geq n^{\frac{1}{2}}.$$

$$= n^{\left(\frac{1}{2}k\right)}.$$

◇

3.2 Die Existenz einer Hierarchie der Zustandsänderungskomplexität

In der Einführung wurde ein ZA angegeben, der mit 3 Zustandsänderungen pro Automat die Sprache $\{ a^n b^n c^n \mid n \in \mathbb{N} \}$ aus $\mathcal{L}_1 \setminus \mathcal{L}_2$ akzeptiert. Somit können auch nichttriviale Sprachen von Zellularautomaten mit konstanter Zustandsänderungsbeschränkung erkannt werden.

Es ist leicht einzusehen, daß bereits eine Zustandsänderung pro Automat ausreicht, um alle regulären Sprachen zu erkennen :

Ein endlicher Automat kann von einem ZA simuliert werden, indem zum Zeitpunkt $t = 0$ am rechten Rand der Eingabe ein Signal gestartet wird, das, während es mit Einheitsgeschwindigkeit nach links läuft, in jedem Schritt ein neues Zeichen der Eingabe liest und seinen Zustand entsprechend der Überföhrungsfunktion des endlichen Automaten ändert. Wird das Signal im Anfangszustand des endlichen Automaten gestartet, so erhält man nach n Schritten das Ergebnis des Erkennungsprozesses im akzeptierenden Automaten am linken Rand der Retina.

Viele komplexere Sprachen können jedoch nur mit wesentlich mehr Zustandsänderungen von Zellularautomaten erkannt werden. Im folgenden wird für einige Sprachen die minimale Anzahl der zu ihrer Erkennung notwendigen Zustandsänderungen untersucht. Mit den Ergebnissen wird dann eine Hierarchie von Sprachfamilien verschiedener Zustandsänderungskomplexität aufgestellt.

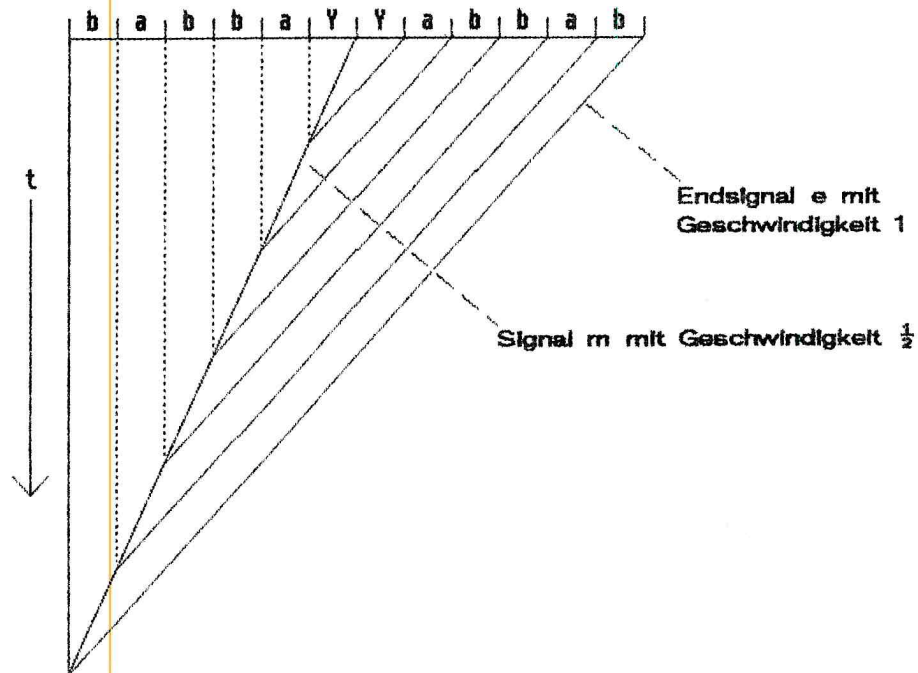
Die Existenz einer solchen Hierarchie wurde für Realzeit-Zellularautomaten bereits in [Vo82b] bewiesen, der Beweis wird hier in modifizierter Form anhand von Realzeitsprachen in unidirektionalen ZA gebracht. Dadurch ist es möglich, den Beweis gleichzeitig für die Sprachklassen der UZA und ZA mit und ohne Realzeitbeschränkung zu führen.

Beispiel 3.1 Es wird die Sprache $L_R = \{ uYYu^R \mid u \in \{a,b\}^+ \}$ betrachtet. L_R ist als lineare Sprache von unidirektionalen Zellularautomaten in Realzeit erkennbar - und damit von $O(n)$ -zustandsänderungsbeschränkten UZA. Ein einfacher Algorithmus für L_R wird hier kurz erläutert und an einem Beispielwort in Abb. 3.1 veranschaulicht.

Funktionsweise: Von der mutmaßlichen Mitte des Eingabewortes, d.h. ausgehend vom Teilwort YY, wird ein Signal m mit der Geschwindigkeit $\frac{1}{2}$ nach links geschickt. Erreicht es zum gleichen Zeitpunkt wie das vom rechten Rand mit Einheitsgeschwindigkeit kommende Signal e die akzeptierende Zelle, so ist die Zeichenkette YY tatsächlich in der Mitte der Eingabe. Außerdem prüft e das Eingabewort auf die Form $\{a,b\}^+YY\{a,b\}^+$.

Gleichzeitig wird für jedes Zeichen der Eingabe aus $\{a,b\}$ ein a - bzw. b -Signal mit Geschwindigkeit 1 nach links geschickt und mit dem Inhalt der Zelle verglichen, in der das langsamere Signal m eingeholt wird. Verlaufen alle Vergleiche positiv, so akzeptiert der UZA die Eingabe.

Abb. 3.1 Algorithmus zum Erkennen der Sprache $L_R = \{uYYu^R \mid u \in \{a,b\}^+\}$ am Beispiel der Eingabe *babbaYYabbab*



In der Zeichnung sind zur Vereinfachung nur die a- und b-Signale aus dem Teilwort u^R eingezeichnet. Da eine Zelle im ersten Schritt der Berechnung noch nicht entscheiden kann, ob sie sich vor oder hinter dem Teilwort YY befindet, muß auch für jedes a und b des Teilwortes u ein Signal nach links geschickt werden. Erreicht ein a- oder b-Signal den linken Rand, ohne die beiden Y passiert zu haben, so wird es gelöscht, ein Vergleich findet nicht statt.

Auch die anderen Algorithmen dieses Kapitels werden entsprechend vereinfacht dargestellt, es werden nur die „wesentlichen“ Signale eingezeichnet. Man kann sich jedoch bei allen dargestellten Verfahren leicht überlegen, daß die Anzahl der benötigten Zustandsänderungen durch diese Vereinfachungen höchstens um einen Faktor verringert wird, die Größenordnung wird nicht verändert.

Es wird nun bewiesen, daß zur Erkennung der Sprache L_R tatsächlich $O(n)$ Zustandsänderungen pro Automat benötigt werden :

Lemma 3.5 Es sei $L_R = \{uYYu^R \mid u \in \{a,b\}^+\}$ und $s, \epsilon \in \mathbb{R}, \epsilon > 0$. Dann gibt es keinen $s n^{1-\epsilon}$ -max-zustandsänderungsbeschränkten Zellularautomaten, der L_R erkennt.
[Vo82b]

Beweis: Die hier verwendete Beweistechnik wird noch in weiteren Beweisen benutzt werden und verläuft in den folgenden Schritten:

- 1) Es wird die Maximalzahl der in einem $sn^{(1-\varepsilon)}$ -max-zustandsänderungsbeschränkten ZA existierenden Verhaltensfolgeausschnitte $\langle c_{\circ}^w(v, v+1) \rangle$ mit $w \in V^+$ ermittelt.
- 2) Durch geeignete Abschätzungen wird gezeigt, daß diese Größe für eine genügend große Wortlänge n echt kleiner ist als die Anzahl der Wörter der Länge n aus L_R .
- 3) Zur Erkennung von L_R wird jedoch für jedes Wort w der Sprache ein eigener Verhaltensfolgeausschnitt $\langle c_{\circ}^w(v, v+1) \rangle$ benötigt. Denn sind w und \bar{w} zwei verschiedene Wörter der Länge $n = 2k+2$, $k \in \mathbb{N}$ aus L_R mit $w = uYYu^R$ und $\bar{w} = \bar{u}YY\bar{u}^R$, so würde nach Lemma 3.2 aus

$$\langle c_{\circ}^w(k+1, k+2) \rangle = \langle c_{\circ}^{\bar{w}}(k+1, k+2) \rangle$$

folgen, daß auch $uYY\bar{u}^R$ akzeptiert wird. Dies stände jedoch im Widerspruch zur Palindromeigenschaft der Wörter aus L_R .

Ein $sn^{(1-\varepsilon)}$ -max-änderungsbeschränkter ZA ist nach Lemma 3.1 $sn^{(2-\varepsilon)}$ -zeitbeschränkt. Die Zahl der Verhaltensfolgeausschnitte $\langle c_{\circ}(v, v+1) \rangle$ in einem $sn^{(1-\varepsilon)}$ -max-änderungsbeschränkten Zellularautomaten ist damit nach Lemma 3.4 b) mit $\rho = sn^{(1-\varepsilon)}$ und $\tau = sn^{(2-\varepsilon)}$ für große n beschränkt durch

$$\begin{aligned} & (sn^{(2-\varepsilon)})^{2sn^{(1-\varepsilon)}} |A|^{2(sn^{(1-\varepsilon)}+1)} \\ < & (sn^2)^{2sn^{(1-\varepsilon)}} (|A|^2)^{2sn^{(1-\varepsilon)}} \quad \text{für } n \text{ mit } sn^{(1-\varepsilon)} \geq 1 \\ & \leq (n^5)^{2sn^{(1-\varepsilon)}} \quad \text{für } n \geq \max\{|A|, s\}. \\ & = 2^{(10sn^{(1-\varepsilon)} \cdot \ln n)} \\ & \leq 2^{\left(\frac{1}{2}n-1\right)} \quad \text{für große } n, \text{ da } n^{-\varepsilon} \cdot \ln n \text{ dann gegen null geht.} \end{aligned}$$

Da es in L_R aber nur $2^k = 2^{\left(\frac{1}{2}n-1\right)}$ Wörter der Länge $n = 2k+2$ gibt, kann L_R von keinem $sn^{(1-\varepsilon)}$ -max-änderungsbeschränkten ZA erkannt werden.

◇

Aus Lemma 3.6 folgt insbesondere, daß L_R von keinem $O(n^{\frac{1}{2}})$ -max-änderungsbeschränkten ZA erkannt wird. Im folgenden werden Sprachen definiert, die $O(n^{\frac{1}{2}})$ bzw. $O(\ln n)$ Zustandsänderungen pro Automat benötigen.

Definition 3.1 a) $Q_{\#}, D_{\#} : V^+ \rightarrow (V \cup \{\#\})^+$ seien definiert durch

$$Q_{\#}(w_1 w_2 \dots w_n) := w_1 \#^{2(n-1)} w_2 \#^{2(n-2)} \dots \#^4 w_{n-1} \#^2 w_n.$$

$$D_{\#}(w_1 w_2 \dots w_n) := w_1 \#^{(2^{n-1}-1)} w_2 \#^{(2^{n-2}-1)} \dots \#^3 w_{n-1} \# w_n \#$$

für $w = w_1 w_2 \dots w_n \in V^+$.

b) $h_{\#} : (V \cup \{\#\})^* \rightarrow V^*$ sei Homomorphismus mit

$$h_{\#}(a) = \begin{cases} a & a \in V \\ \lambda & a = \# \end{cases}$$

$$c) L_{RQ} := \{vXXwYYQ_{\#}(\$^n)ZZ\$^{n\#^t} \mid h_{\#}(v) = u \in \{a,b\}^+ \wedge w = v^R \wedge |u| = n \wedge |v| = n^2 \wedge t = 3n^2 + n + 6\},$$

$$L_{RD} := \{vXXwYYD_{\#}(\$^n)ZZ\$^{n\#^t} \mid h_{\#}(v) = u \in \{a,b\}^+ \wedge w = v^R \wedge |u| = n \wedge |v| = 2^n \wedge t = 3 \cdot 2^n + n + 6\}.$$

Durch die Abbildungen $D_{\#}$ und $Q_{\#}$ wird ein Wort der Länge n durch Auffüllen mit $\#$ - Zeichen auf die Länge 2^n bzw. n^2 gebracht, zum Beispiel

$$\begin{aligned} abcba &\xrightarrow{D_{\#}} a \text{#####} b \text{#####} c \text{###} b \# \# a \# \\ abcba &\xrightarrow{Q_{\#}} a \text{#####} b \text{#####} c \text{#####} b \# \# a. \end{aligned}$$

Für die Länge der Wörter

$$w_Q := v_1 X X w_1 Y Y Q_{\#}(\$^n) Z Z \$^{n\#^t} \in L_{RQ}$$

und

$$w_D := v_2 X X w_2 Y Y D_{\#}(\$^n) Z Z \$^{n\#^t} \in L_{RD}$$

folgt daraus :

$$|w_Q| = 6n^2 + 2n + 12 = O(n^2) \quad \text{und} \quad |w_D| = 6 \cdot 2^n + 2n + 12 = O(2^n).$$

Das Wort $w = u Y Y u^R$ mit $|u| = n$ aus L_R hat dagegen nur die Länge $2n + 2 = O(n)$.

Die Zahl der nötigen Zustandsänderungen pro Automat ist, wie in Lemma 3.7 bewiesen wird, bei der Erkennung eines Wortes w_Q aus L_{RQ} oder w_D aus L_{RD} nicht wesentlich größer als bei der Erkennung des „entsprechenden“ Wortes w aus L_R . Da die Zustandsänderungskomplexität einer Sprache in Abhängigkeit von der Wortlänge n definiert ist, sind L_{RQ} und L_{RD} bereits von einem $O(n^2)$ - bzw. $O(\ln n)$ -änderungsbeschränkten ZA erkennbar.

In Lemma 3.8 wird schließlich gezeigt, daß diese Zustandsänderungszahlen tatsächlich benötigt werden; damit ist dann die Existenz einer Hierarchie der Zustandsänderungskomplexität bewiesen.

Die in Definition 3.1 c) definierten Sprachen L_{RQ} und L_{RD} sehen auf den ersten Blick ziemlich unübersichtlich aus. Dies ist jedoch durch die an die Sprachen gestellten Anforderungen bedingt :

Die Sprachen sollen bereits von einem unidirektionalen ZA in Realzeit erkennbar sein. So besteht der Zweck des letzten Teilwortes $\#^t$ einzig und allein darin, die Wörter der Sprache so zu verlängern, daß eine Erkennung in Realzeit möglich wird. Die Schwierigkeit bei der Definition geeigneter Sprachen und dem Entwurf der entsprechenden Erkennungsverfahren besteht nicht zuletzt darin, daß auch bei Eingabewörtern aus dem Komplement der Sprachen in keiner Zelle mehr als $O(n^{\frac{1}{2}})$ bzw. $O(\ln n)$ Zustandsänderungen stattfinden dürfen.

Es ist sicherlich möglich, die Hierarchie der Zustandsänderungskomplexität anhand einfacherer Beispiele zu zeigen. Der Vorteil der hier definierten Sprachen L_{RQ} und L_{RD} ist jedoch, daß man mit ihnen die Hierarchie für ZA und UZA mit und ohne Realzeitbeschränkung durch einen einzigen Beweis zeigen kann. Außerdem wird im nächsten Abschnitt anhand sehr ähnlicher Sprachen die aufgestellte Hierarchie verfeinert, die dort verwendeten und noch etwas komplizierteren Algorithmen werden dadurch etwas verständlicher.

Im folgenden Beispiel wird versucht, die Funktion der Abbildungen $Q_{\#}$ und $D_{\#}$ und damit die Struktur der Sprachen L_{RQ} und L_{RD} zu verdeutlichen.

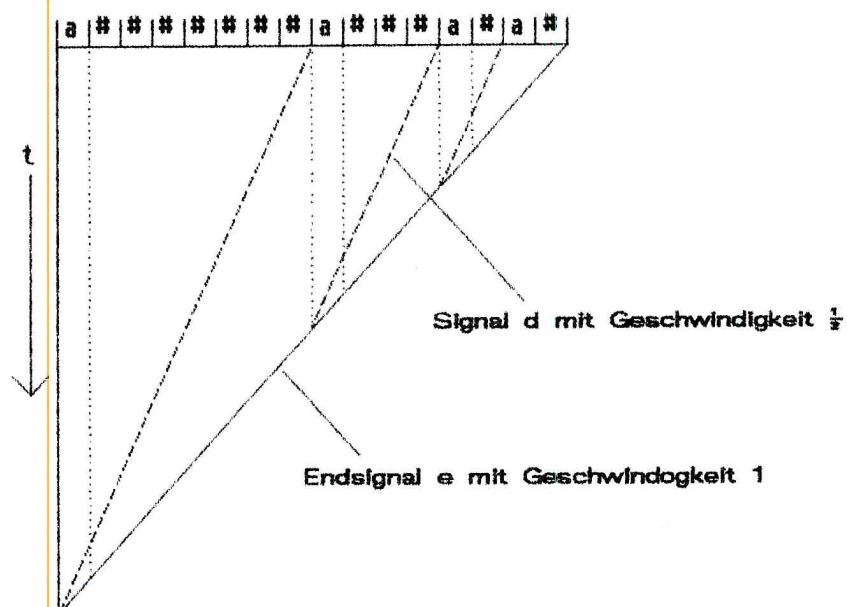
Beispiel 3.2 Es werden die Sprachen $L_D := D_{\#}(\{a\}^+)$ und $L_Q := Q_{\#}(\{a\}^+)$ betrachtet. Für $D_{\#}$ und $Q_{\#}$ gilt nach obiger Bemerkung:

$$|w| = n \implies |D_{\#}(w)| = 2^n \wedge |Q_{\#}(w)| = n^2$$

L_D und L_Q ähneln also den unitären Sprachen $\{\#^{2^n} | n \in \mathbb{N}\}$ und $\{\#^{n^2} | n \in \mathbb{N}\}$, sie sind im Gegensatz zu diesen aber noch an „geeigneten“ Stellen durch a's markiert. Mit Hilfe dieser Markierungen können L_D und L_Q von unidirektionalen ZA in Realzeit erkannt werden.

a) Erkennungsverfahren für die Sprache L_D :

Abb. 3.2 Erkennung von L_D am Beispiel der Eingabe $D_{\#}(a^4)$



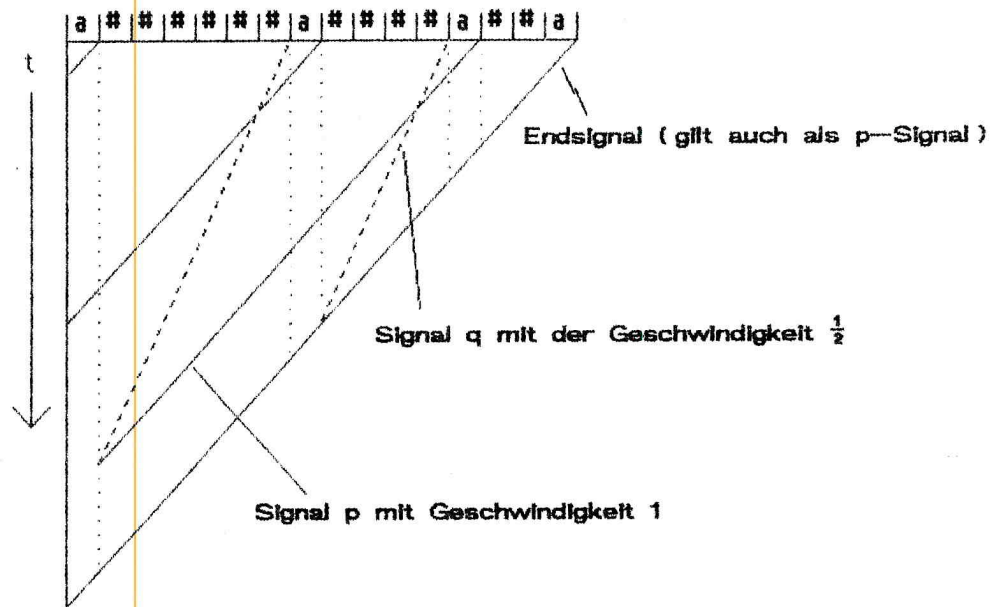
Funktionsweise: Die Wörter der Sprache L_D haben die Eigenschaft, daß die Entfernung von einem Zeichen a zum nächsten links von ihm um eins größer ist als die Entfernung zum rechten Rand.

Somit kann ein ZA L_D erkennen, indem von jeder mit a markierten Zelle aus ein Signal d mit Geschwindigkeit $\frac{1}{2}$ nach links geschickt wird. Werden diese Signale direkt nach der jeweils nächsten mit a markierten Zelle vom Signal e , das vom rechten Rand mit Einheitsgeschwindigkeit nach links läuft, eingeholt, dann hat die Eingabe die Form $D_{\#}^+(w), w \in \{a\}^+$.

Zur Erkennung von L_D werden so nicht mehr als drei Zustandsänderungen pro Automat benötigt.

b) Erkennungsverfahren für die Sprache L_Q :

Abb. 3.3 Erkennung von L_Q am Beispiel der Eingabe $Q_{\#}(a^4)$



Funktionsweise: Die Wörter aus L_Q sind dadurch gekennzeichnet, daß direkt links von jedem a zwei $\#$ mehr stehen als direkt rechts davon.

Im oben skizzierten Erkennungsverfahren wird diese Eigenschaft überprüft, indem von jedem a aus ein Signal q mit der Geschwindigkeit $\frac{1}{2}$ nach links gesendet wird; von der Zelle rechts daneben startet ein Signal p mit Einheitsgeschwindigkeit in die gleiche Richtung. Wird jedes q -Signal direkt vor der jeweils nächsten mit a markierten Zelle zum zweiten Mal von einem p -Signal überholt, und zählt das Endsignal direkt links vom letzten a genau zwei $\#$ -Zeichen, so ist die Eingabe aus L_Q .

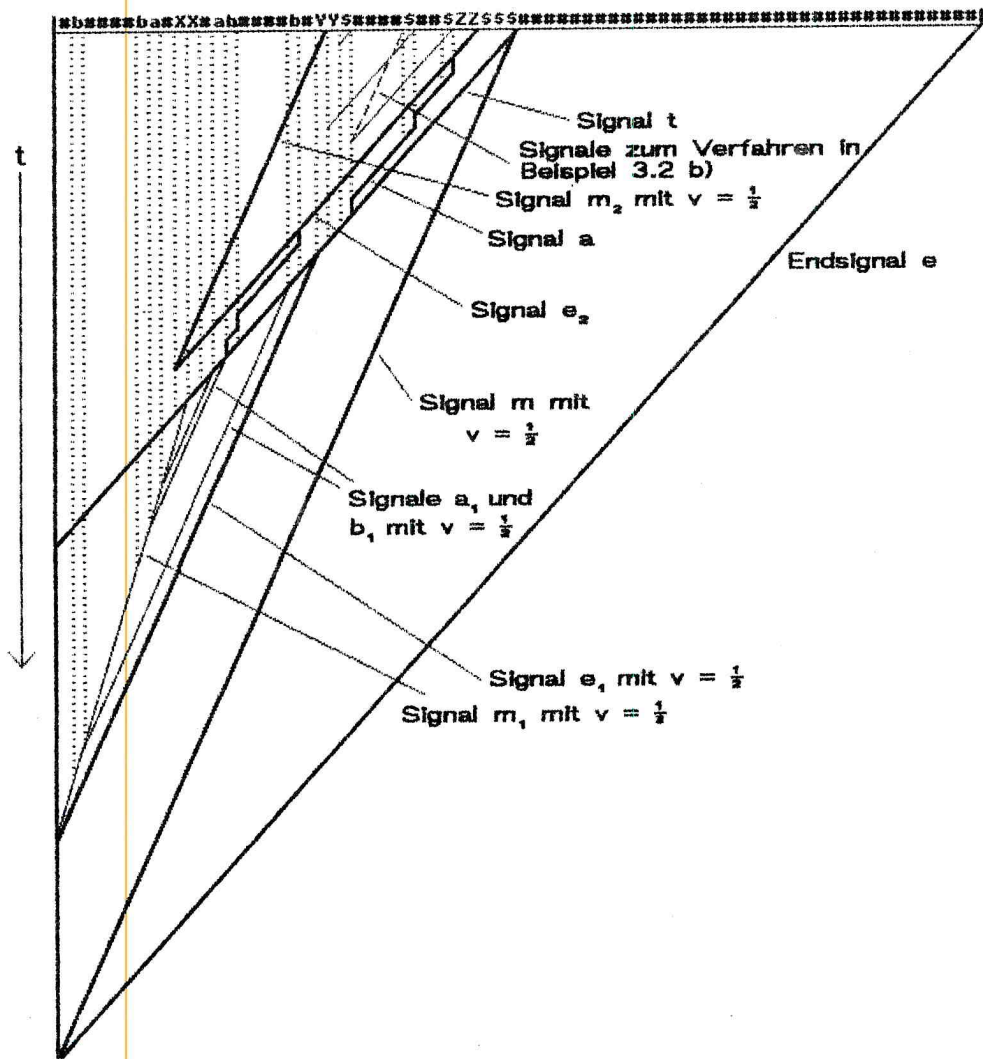
Der ZA benötigt dabei maximal 5 Zustandsänderungen pro Automat.

Mit Hilfe der Abbildungen $D_{\#}$ und $Q_{\#}$ ist es also möglich, Wörter der Länge 2^n bzw. n^2 zu erzeugen. Umgekehrt können die Erkennungsverfahren des letzten Beispiels verwendet werden, um geeignet strukturierte Wörter auf die Länge 2^n bzw. n^2 zu prüfen. Die Teilwörter v und w in den Sprachen L_{RQ} und L_{RD} haben diese Länge, es sind aber jeweils nur n Zeichen aus $\{a,b\}$ in v und in w , der Rest besteht aus $\#$ -Zeichen. Das Teilwort $vXXw$ ist also ein Palindrom, bei dem im Gegensatz zu einem Wort aus der Sprache L_R aus Beispiel 3.1 die Dichte der a - und b -Zeichen stark verringert worden ist. Dadurch wird, wie im folgenden Lemma gezeigt wird, auch die Anzahl der benötigten Zustandsänderungen pro Zelle entsprechend verringert.

- Lemma 3.7** a) $L_{RQ} \in \mathcal{A}UZAR(O(n^{\frac{1}{2}}))$
 b) $L_{RD} \in \mathcal{A}UZAR(O(\ln n))$

Beweis: a) Ein Algorithmus für L_{RQ} wird anhand der folgenden Zeichnung erläutert :

Abb. 3.4 Erkennung von L_{RQ} am Beispiel der Eingabe $vXXwYYQ_{\#}(\$^3)ZZ\$^{3\#t}$ mit $t = 36$



Funktionsweise: Das skizzierte Erkennungsverfahren besteht aus zwei Phasen, die durch das Signal t voneinander getrennt werden. In der ersten Phase oberhalb von t wird die Eingabe auf Wohlgeformtheit geprüft, insbesondere wird geprüft, ob die Eingabe die richtige Länge hat. In dieser Phase wird nur eine konstante Anzahl von Zustandsänderungen pro Zelle benötigt. Erst in der zweiten Phase wird die Eigenschaft $w = v^R$ überprüft, dazu werden $O(n^{\frac{3}{2}})$ Zustandsänderungen gebraucht. Durch diese Einteilung in Phasen wird sichergestellt, daß auch bei Eingabewörtern aus dem Komplement von L_{RQ} in keiner Zelle mehr als $O(n^{\frac{3}{2}})$ Zustandsänderungen stattfinden.

In der ersten Phase laufen die folgenden Prozesse gleichzeitig nebeneinander ab:

- 1) Der zwischen YY und ZZ liegende Teil der Eingabe wird durch das in Beispiel 3.2 b) beschriebene Verfahren auf die Form $Q_{\#}(\$^k)$ überprüft, die dazu benötigten Signale p und q sind in dünnen Linien eingezeichnet.
- 2) Das Teilwort $Q_{\#}(\$^k)ZZ\n wird auf die Bedingung $k=n$ getestet. Dazu wird das Signal a , das am linken Rand von $\n mit Geschwindigkeit 1 gestartet wird, an jedem $\$$ -Zeichen in $Q_{\#}(\$^k)$ um einen Takt verzögert. Wird es dabei im am weitesten links stehenden $\$$ vom Signal t , das am rechten Rand der $\n gestartet wird, eingeholt, so gilt $k=n$.
- 3) Mit Hilfe der Signale m_2 und e_2 werden die Teilwörter w und $Q_{\#}(\$^n)$ auf gleiche Länge getestet.
- 4) Mit derselben Technik wie in 2) wird geprüft, ob im Teilwort w genau n Zeichen aus $\{a,b\}$ sind. Auch hier wird ein a -Signal an jedem Zeichen aus $\{a,b\}$ um einen Zeittakt verzögert, außerdem werden die angetroffenen a 's und b 's markiert.

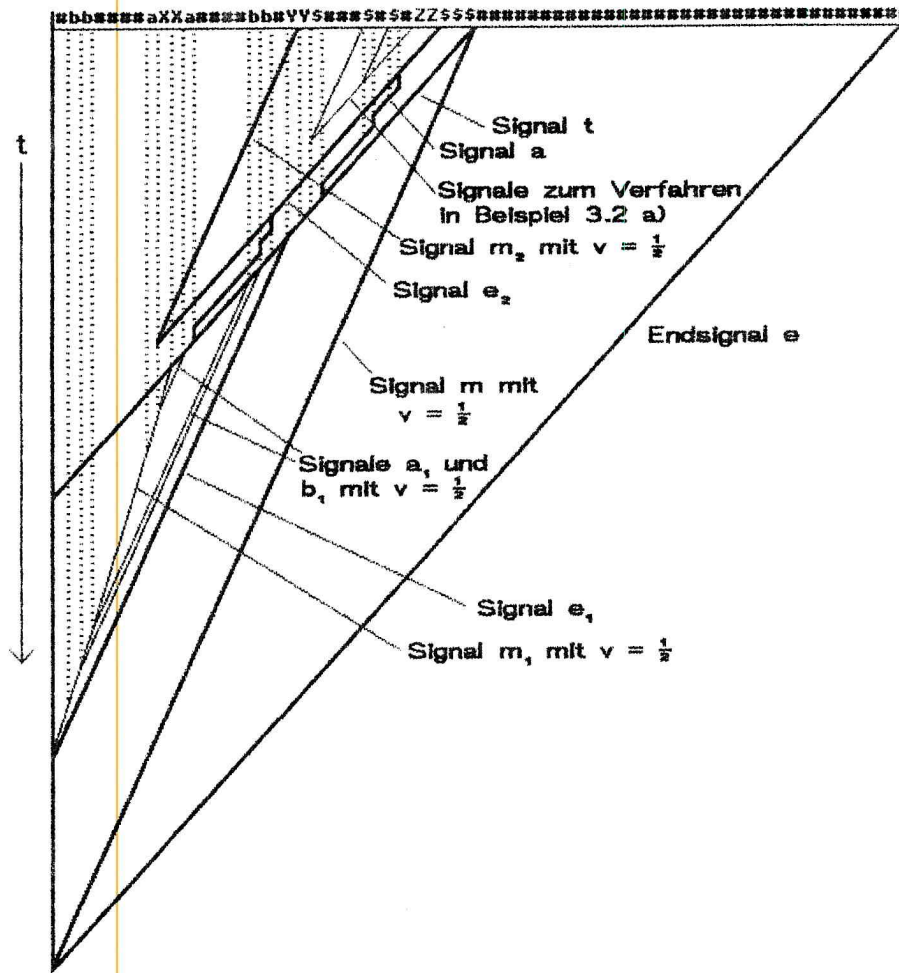
In der zweiten Phase bleibt noch zu überprüfen, ob für das Teilwort w die Bedingung $w = v^R$ gilt. Dazu wird von jedem markierten a und b des Teilwortes w aus ein a_1 - bzw. b_1 -Signal mit Geschwindigkeit $\frac{1}{2}$ nach links geschickt. Die a_1 - und b_1 -Signale werden jeweils durch das Eintreffen des t -Signals gestartet. Trifft das t -Signal auf das Teilwort XX , so wird außerdem ein Signal m_1 gestartet, das mit Geschwindigkeit $\frac{1}{2}$ nach links läuft. Wenn jedes a_1 - bzw. b_1 -Signal das Signal m_1 in einer Zelle einholt, die wieder ein a bzw. b enthält, so folgt $w = v^R$.

Mit den Signalen m und e wird geprüft, ob am Ende der Eingabe die richtige Anzahl von $\#$ -Zeichen steht.

Da für alle Prozesse in der ersten Phase nur eine konstante Anzahl von Zustandsänderungen pro Zelle benötigt wird, ist die Anzahl der Zustandsänderungen bestimmt durch die Anzahl der a_1 - und b_1 -Signale und damit durch die Anzahl der Zeichen aus $\{a,b\}$ im Eingabewort. Für die Erkennung einer Eingabe der Länge $n = 6k^2 + 2k + 12$ aus L_{RQ} werden daher in keiner Zelle mehr als $O(k) = O(n^{\frac{3}{2}})$ Zustandsänderungen benötigt. Da in der zweiten Phase der Erkennung nur von den markierten Zeichen aus ein a_1 - bzw. b_1 -Signal gestartet wird, ist sichergestellt, daß auch bei Eingabe eines Wortes aus dem Komplement von L_{RQ} nicht wesentlich mehr Zustandsänderungen stattfinden.

b) Ein Erkennungsverfahren für die Sprache L_{RD} :

Abb. 3.5 Erkennung von L_{RD} am Beispiel der Eingabe $vXXwYYD_{\#}(\$^3)ZZ\$^3_{\#}t$ mit $t = 33$



Funktionsweise: Der Algorithmus für die Sprache L_{RD} verläuft analog zu a). Es wird lediglich in der ersten Phase in 2) der Algorithmus für $Q_{\#}(w)$ aus Beispiel 3.2 b) durch den für $D_{\#}(w)$ aus Beispiel 3.2 a) ersetzt.

Auch in diesem Fall ist die Anzahl der Zustandsänderungen bestimmt durch die Anzahl der Zeichen aus $\{a,b\}$ im Eingabewort. Für die Erkennung einer Eingabe der Länge $n = 6 \cdot 2^k + 2k + 12$ aus L_{RD} werden in keiner Zelle mehr als $O(k) = O(\ln n)$ Zustandsänderungen benötigt.

◇

Im nächsten Lemma wird gezeigt, daß eine wesentliche Verringerung der Anzahl der Zustandsänderungen pro Zelle gegenüber den eben beschriebenen Algorithmen auch in bidirektionalen ZA ohne Realzeitbeschränkung nicht möglich ist.

Lemma 3.8 a) Es seien $\epsilon, s \in \mathbb{R}$ mit $\epsilon > 0$. Dann gibt es keinen $sn^{(\frac{1}{2}-\epsilon)}$ -max-änderungsbeschränkten ZA, der L_{RQ} erkennt.

b) Es sei $s \in \mathbb{R}$. Dann gibt es keinen $s \ln(\ln n)$ -max-änderungsbeschränkten ZA, der L_{RD} erkennt.

Beweis: Die Beweise laufen beide analog zum Beweis von Lemma 3.6 ab : Es wird bewiesen, daß die Maximalzahl der in einem zustandsänderungsbeschränkten ZA existierenden Verhaltensfolgeausschnitte $\langle c_{\circ}^w(v, v+1) \rangle$ kleiner ist als die Anzahl der Wörter einer Länge n aus L_{RQ} bzw. L_{RD} , sofern n nur groß genug gewählt wird.

a) Wie bei der Erkennung von L_R wird auch für jedes Wort w aus der Sprache L_{RQ} ein eigener Verhaltensfolgeausschnitt $\langle c_{\circ}^w(v, v+1) \rangle$ benötigt. Denn sind $w_Q = vXXwYYQ_{\#}(\$^k)ZZ\$^{k\#t}$ und $\bar{w}_Q = \bar{v}XX\bar{w}YYQ_{\#}(\$^k)ZZ\$^{k\#t}$ zwei verschiedene Wörter der Länge $n = 6k^2 + 2k + 12$, $k \in \mathbb{N}$ aus L_{RQ} , so würde nach Lemma 3.2 aus

$$\langle c_{\circ}^{w_Q}(k^2+1, k^2+2) \rangle = \langle c_{\circ}^{\bar{w}_Q}(k^2+1, k^2+2) \rangle$$

folgen, daß auch $vXX\bar{w}YYQ_{\#}(\$^k)ZZ\$^{k\#t}$ akzeptiert wird. Hierbei sind k^2+1 und k^2+2 die Nummern der Zellen, die die beiden X enthalten. Wegen $\bar{w} \neq v^R$ ist $vXX\bar{w}YYQ_{\#}(\$^k)ZZ\$^{k\#t}$ aber nicht in L_{RQ} .

Ein $sn^{(\frac{1}{2}-\epsilon)}$ -max-änderungsbeschränkter ZA ist nach Lemma 3.1 $sn^{(\frac{3}{2}-\epsilon)}$ -zeitbeschränkt. Die Maximalzahl der Verhaltensfolgeausschnitte $\langle c_{\circ}^w(v, v+1) \rangle$ ist damit nach Lemma 3.4 b) mit $\rho = sn^{(\frac{1}{2}-\epsilon)}$ und $\tau = sn^{(\frac{3}{2}-\epsilon)}$ für große n beschränkt durch

$$\begin{aligned} & \left(sn^{(\frac{3}{2}-\epsilon)} \right)^{2sn^{(\frac{1}{2}-\epsilon)}} |A|^{2(sn^{(\frac{1}{2}-\epsilon)}+1)} \\ < & \left(sn^2 \right)^{2sn^{(\frac{1}{2}-\epsilon)}} \left(|A|^2 \right)^{2sn^{(\frac{1}{2}-\epsilon)}} \quad \text{für } n \text{ mit } sn^{(\frac{1}{2}-\epsilon)} \geq 1 \\ \leq & \left(n^5 \right)^{2sn^{(\frac{1}{2}-\epsilon)}} \quad \text{für } n \geq \max\{|A|, s\} \\ = & 2^{(10sn^{(\frac{1}{2}-\epsilon)} \cdot \ln n)} \\ \leq & 2^{\left(\frac{1}{7}n\right)^{\frac{1}{2}}} \quad \text{für große } n, \text{ da } n^{-\epsilon} \cdot \ln n \text{ dann gegen null geht.} \end{aligned}$$

Zu jedem Wort v der Länge k^2 mit k Zeichen aus $\{a, b\}$ gibt es genau ein Wort $vXXvRYYQ_{\#}(\$^k)ZZ\$^k\#^t$ der Länge $n = 6k^2 + 2k + 12$ in L_{RQ} . Es gibt

$$\binom{k^2}{k} 2^k$$

solche Wörter v , dies ist folglich auch die Anzahl der Wörter der Länge n in L_{RQ} . Wegen $n = 6k^2 + 2k + 12$ gilt $(\frac{1}{7}n)^{\frac{1}{2}} < k$ für große n . Daraus folgt

$$2^{(\frac{1}{7}n)^{\frac{1}{2}}} < 2^k < \binom{k^2}{k} 2^k.$$

Also kann L_{RQ} von keinem $sn^{(\frac{1}{2}-\epsilon)}$ -max-änderungsbeschränkten ZA erkannt werden.

- b) Auch im Fall von L_{RD} wird für jedes Wort w einer Länge $n = 6 \cdot 2^k + 2k + 12$ aus der Sprache ein eigener Verhaltensfolgeausschnitt $\langle c_o(v, v+1) \rangle$ benötigt. Zum Beweis betrachtet man analog zu a) Lemma 3.2 mit $v = 2^{k+1}$.

Die Zahl der Verhaltensfolgeausschnitte $\langle c_o(v, v+1) \rangle$ in einem $s \ln(\ln n)$ -max-änderungsbeschränkten ZA ist nach Lemma 3.1 und Lemma 3.4 b) beschränkt durch

$$\begin{aligned} & (s \ln(\ln n))^{2s \ln(\ln n)} |A|^{2(s \ln(\ln n) + 1)} \\ < & (s \ln(\ln n))^{2s \ln(\ln n)} (|A|^2)^{2s \ln(\ln n)} \quad \text{für } n \text{ mit } s \ln(\ln n) > 1 \\ < & (n^5)^{2s \ln(\ln n)} \quad \text{für } n \geq \max\{|A|, s\}, \text{ außerdem wird } \ln(\ln n) \text{ in der} \\ & \quad \text{Basis durch } n \text{ abgeschätzt.} \\ = & n^{10s \ln(\ln n)} \\ < & \left(\frac{1}{7}n\right)^{\frac{1}{2} \text{ld}\left(\frac{1}{7}n\right)} \quad \text{für große } n. \end{aligned}$$

Zu jedem Wort v der Länge 2^k mit k Zeichen aus $\{a, b\}$ gibt es genau ein Wort $vXXvRYYD_{\#}(\$^k)ZZ\$^k\#^t$ der Länge $n = 6 \cdot 2^k + 2k + 12$ in L_{RD} . Es gibt

$$\binom{2^k}{k} 2^k$$

solche Wörter v , dies ist folglich auch die Anzahl der Wörter der Länge n in L_{RD} . Wegen $n = 6 \cdot 2^k + 2k + 12$ gilt $\text{ld}\left(\frac{1}{7}n\right) < k$ für große n . Daraus folgt

$$\left(\frac{1}{7}n\right)^{\frac{1}{2}} \text{ld}\left(\frac{1}{7}n\right) < (2^k)^{\frac{k}{2}}$$
$$< \binom{2^k}{k} < \binom{2^k}{k} 2^k \quad \text{nach Lemma 3.5.}$$

Also kann L_{RQ} von keinem $s \ln(\ln n)$ -max-änderungsbeschränkten ZA erkannt werden.



Aus Lemma 3.8 folgt insbesondere, daß L_{RQ} von keinem $O(\ln n)$ -max-änderungsbeschränkten und L_{RD} von keinem $O(1)$ -max-änderungsbeschränkten ZA erkannt wird. Aus den Lemmata 3.6, 3.7 und 3.8 folgt nun das Hauptergebnis dieses Abschnitts:

Satz 3.1 Für $Z \in \{ \dot{A}ZA, \dot{A}ZAR, \dot{A}UZA, \dot{A}UZAR \}$ gilt

$$Z(O(1)) \subset Z(O(\ln n)) \subset Z(O(n^{\frac{1}{2}})) \subset Z(O(n))$$

Beweis: In den Algorithmen von Beispiel 3.1 und Lemma 3.7 wurde jeweils ein unidirektionaler, auf Realzeit beschränkter Zellularautomat zur Erkennung der Sprachen L_R , L_{RQ} und L_{RD} verwendet. In den Lemmata 3.6 und 3.8 wurde jedoch bewiesen, daß auch mit einem bidirektionalen ZA ohne Zeitbeschränkung keine kleinere Zustandsänderungszahl zur Erkennung dieser Sprachen erreichbar ist. Damit ist die Existenz einer Hierarchie der Zustandsänderungskomplexität für alle vier in dieser Arbeit betrachteten Klassen änderungsbeschränkter Zellularautomaten gezeigt.



3.3 Verfeinerungen der Hierarchie

Im vorhergehenden Abschnitt wurde die Existenz einer Hierarchie der Zustandsänderungskomplexitäten $O(n)$, $O(n^{\frac{1}{2}})$, $O(\ln n)$ und $O(1)$ bewiesen. Im folgenden wird gezeigt, daß es zwischen diesen Klassen weitere, unendliche Hierarchien der Zustandsänderungskomplexität gibt. Es werden Sprachen vorgestellt, zu deren Erkennung Zustandsänderungszahlen von $O(\ln n)$, $O(\ln(\ln n))$, $O(\ln(\ln(\ln n)))$, ... sowie von $O(n^{\frac{1}{2}})$, $O(n^{\frac{1}{4}})$, $O(n^{\frac{1}{8}})$, ... benötigt werden. Die Konstruktion dieser Sprachen erfolgt wie bei den Sprachen L_{RQ} und L_{RD} aus dem letzten Abschnitt mit Hilfe der Abbildungen $Q_{\#}$ und $D_{\#}$.

Definition 3.2 a) $Q_{(k)}, D_{(k)} : V^+ \rightarrow (V \cup \{\#, \$, Z\})^+$ seien definiert durch

$$Q_{(k)}(w) := v_k Z v_{k-1} Z \dots v_2 Z v_1 \text{ mit } v_{n+1} = \begin{cases} Q_{\#}(\$^{|v_n|}) & n > 0 \\ Q_{\#}(w) & n = 0 \end{cases}$$

$$D_{(k)}(w) := v_k Z v_{k-1} Z \dots v_2 Z v_1 \text{ mit } v_{n+1} = \begin{cases} D_{\#}(\$^{|v_n|}) & n > 0 \\ D_{\#}(w) & n = 0 \end{cases}$$

$$b) \ln_{(k)} n := \begin{cases} \ln(\ln_{(k-1)} n) & k > 1 \\ \ln n & k = 1 \end{cases}$$

$$2_{(k)}^n := \begin{cases} 2^{(2_{(k-1)}^n)} & k > 1 \\ 2^n & k = 1 \end{cases}$$

$$c) L_{RQ_{(k)}} := \{vXXwYYQ_{(k)}(\$^n)ZZ\$^{n\#} \mid h_{\#}(v) = u \in \{a,b\}^+ \wedge w = v^R \\ \wedge |u| = n \wedge |v| = n^{(2^k)} \wedge t = |vXXwYYQ_{(k)}(\$^n)ZZ\$^{n\#}| \},$$

$$L_{RD_{(k)}} := \{vXXwYYD_{(k)}(\$^n)ZZ\$^{n\#} \mid h_{\#}(v) = u \in \{a,b\}^+ \wedge w = v^R \\ \wedge |u| = n \wedge |v| = 2_{(k)}^n \wedge t = |vXXwYYD_{(k)}(\$^n)ZZ\$^{n\#}| \},$$

Für ein Wort $Q_{(k)}(w) = v_k Z \dots Z v_1$ gilt aufgrund der Definition von $Q_{(k)}$ mit Hilfe der Abbildung $Q_{\#}$:

$$|v_{n+1}| = \begin{cases} |v_n|^2 & n > 0 \\ |w|^2 & n = 0 \end{cases}$$

Daraus folgt für die Länge des ersten Teilworts von $Q_{(k)}(w)$ $|v_k| = |w|^{2^k}$ und für die Gesamtlänge $|Q_{(k)}(w)| = O(|w|^{2^k})$.

Für $D_{(k)}(w)$ folgt analog $|v_k| = 2_{(k)}^{|w|}$ und $|D_{(k)}(w)| = O(2_{(k)}^{|w|})$.

Beispiel 3.3 Die Sprachen $L_{Q_{(k)}} := Q_{(k)}(\{a\}^+)$ und $L_{D_{(k)}} := D_{(k)}(\{a\}^+)$ sind von einem $O(1)$ -änderungsbeschränkten UZA in Realzeit erkennbar. Ein mögliches Erkennungsverfahren für $L_{Q_{(k)}}$ und $L_{D_{(k)}}$ wird hier für $k=2$ vorgestellt.