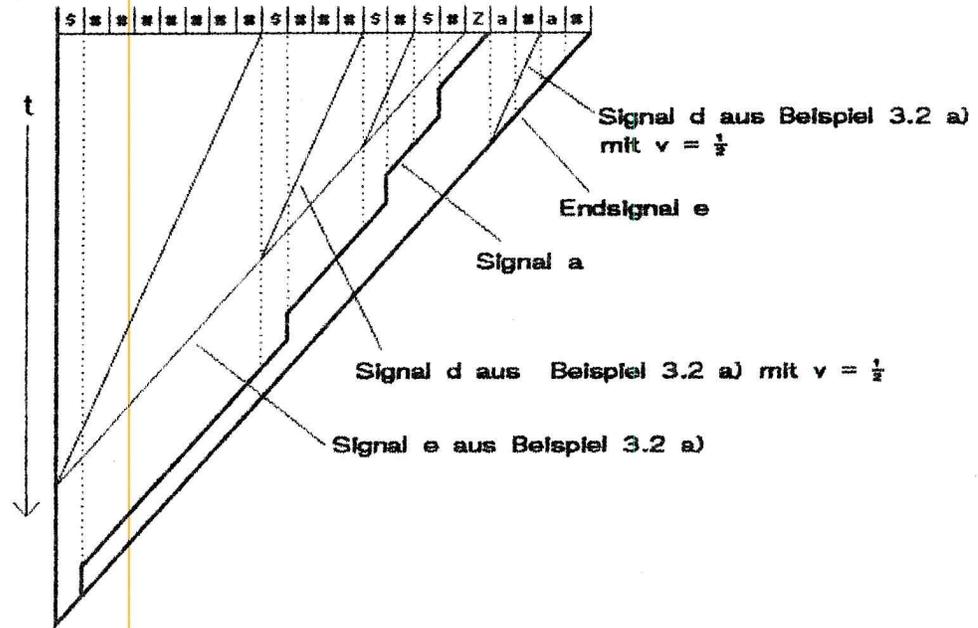




b) Erkennungsverfahren für die Sprache  $L_{D(2)}$ :

Abb. 3.7 Erkennung von  $L_{D(2)}$  am Beispiel des Eingabewortes  $D_{(2)}(aa) = \$*****\$***\$*\$*Za#a* = v_2 Z v_1$



Funktionsweise: Die Erkennung von  $L_{D(2)}$  verläuft ähnlich wie in a) in zwei Abschnitten:

- 1) Das Teilwort  $v_1$  wird auf die Form  $D_{\#}(w), w \in \{a\}^+$  geprüft,  $v_2$  auf die Form  $D_{\#}(w), w \in \{\$\}^+$ . Dazu wird der in Beispiel 3.2 a) beschriebene Algorithmus zur Erkennung von  $D_{\#}(V^+)$  verwendet, die zugehörigen Signale e und d sind in dünnen Linien eingezeichnet ( das Endsignal e in  $v_1$  wird auch in 2) verwendet und ist deshalb dick eingezeichnet ).
- 2) Wie in a) wird mit Hilfe eines a-Signals, das bei jedem  $\$$ -Zeichen einen Takt lang angehalten wird, überprüft, ob es in  $v_2$  genau  $|v_1|$   $\$$ -Zeichen gibt.

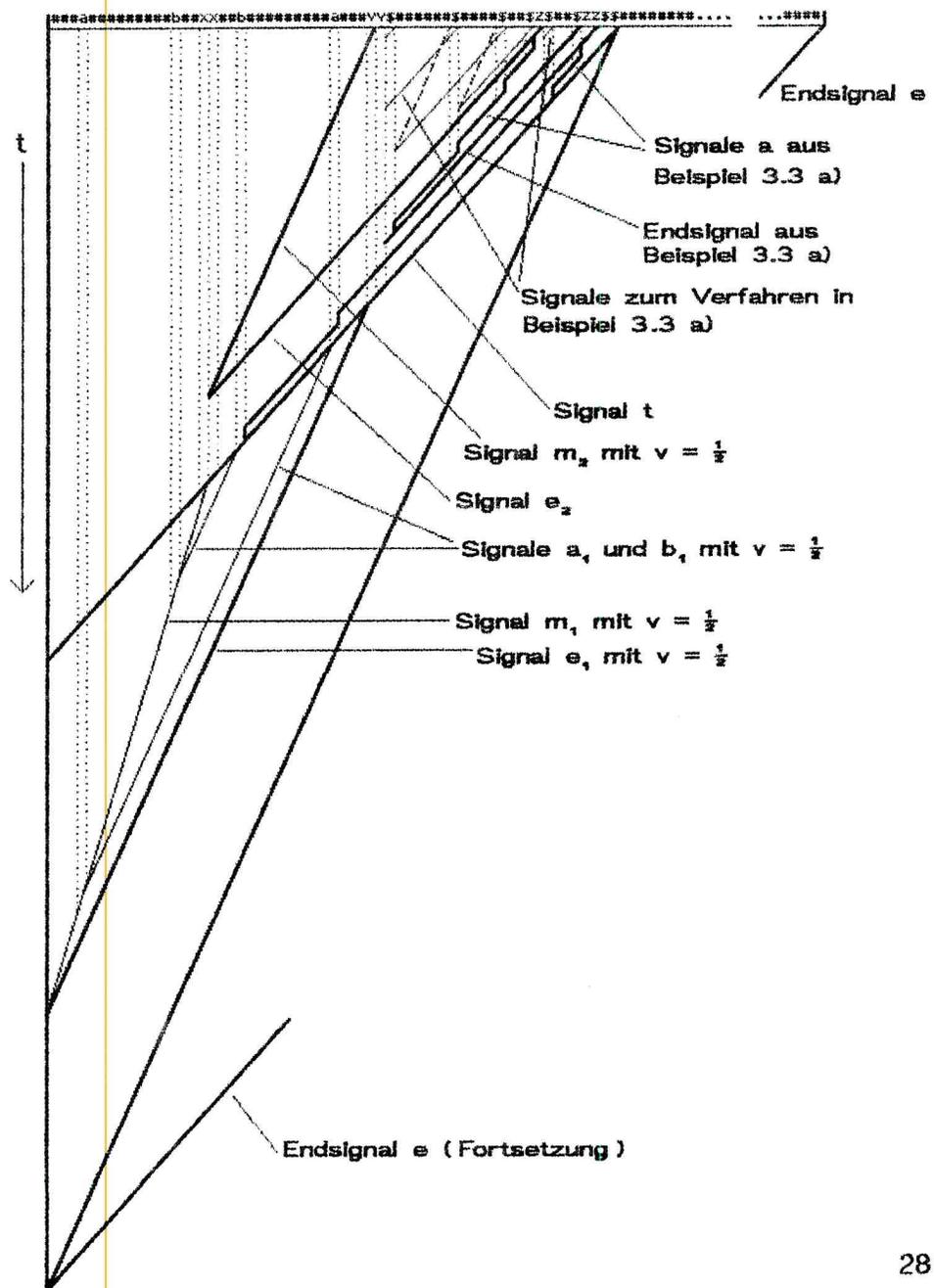
Zur Erkennung von  $L_{D(k)}$  müssen in 1)  $v_2$  bis  $v_k$  auf die Form  $D_{\#}(w), w \in \{\$\}^+$  überprüft werden. Außerdem wird analog zu 2) geprüft, ob für alle  $n$  aus  $\{1, \dots, k-1\}$  genau  $|v_n|$   $\$$ -Zeichen in  $v_{n+1}$  sind. Dabei wird wiederum nur eine konstante Anzahl von Zustandsänderungen pro Automat benötigt.

Unter Verwendung der Verfahren aus dem letzten Beispiel wird nun gezeigt, daß die Sprache  $L_{RQ(k)}$  von einem  $O(n^{\frac{1}{k}})$ -änderungsbeschränkten und  $L_{RD(k)}$  von einem  $O(\ln_{(k)} n)$ -änderungsbeschränkten UZA in Realzeit erkannt wird. Die Erkennungsverfahren sind denen für die Sprachen  $L_{RQ}$  und  $L_{RD}$  aus Lemma 3.7 ähnlich. Insbesondere gilt  $L_{RQ(1)} = L_{RQ}$  und  $L_{RD(1)} = L_{RD}$ ; für diesen Fall sind die Verfahren mit denen aus Lemma 3.7 identisch.

- Lemma 3.9 a)  $L_{RQ(k)} \in \text{ÄUZAR}(O(n^{\frac{1}{k}}))$   
 b)  $L_{RD(k)} \in \text{ÄUZAR}(O(\ln(k)n))$

Beweis: Der Beweis wird ausführlich für das Beispiel  $L_{RQ(2)}$  geführt. Ein Verfahren für  $L_{RQ(n)}$ ,  $n > 2$  sowie für  $L_{RD(n)}$  ergibt sich, indem man den verwendeten Teilalgorithmus für  $L_{Q(2)}$  aus Beispiel 3.3 a) ersetzt durch den für  $L_{Q(n)}$  bzw.  $L_{D(n)}$ .

Abb. 3.8 Erkennung der Sprache  $L_{RQ(2)}$  mit  $O(n^{\frac{1}{2}})$  Zustandsänderungen am Beispiel der Eingabe  $w_Q = vXXwYYQ_{(2)}(\$^2)ZZ\$^{2\#t} = vXXwYYv_2Zv_1ZZ\$^{2\#t}$  mit  $v = \#\#a\#\#\#\#\#\#\#\#\#\#\#\#b\#\#$



Funktionsweise: Der Algorithmus läuft wie die Verfahren für  $L_{RQ}$  und  $L_{RD}$  aus Lemma 3.6 in zwei Phasen ab.

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

- 1) Das Teilwort  $v_2 Z v_1$  wird durch das in Beispiel 3.3 a) beschriebene Verfahren auf die Form  $Q_{(2)}(\$^k)$  überprüft.
- 2) Das Teilwort  $v_2 Z v_1 Z Z \$^n = Q_{(2)}(\$^k) Z Z \$^n$  wird auf die Bedingung  $k=n$  getestet. Dies geschieht wieder durch ein a-Signal, daß durch jedes in  $v_1$  angetroffene  $\$$  einen Takt angehalten wird.
- 3) Mit Hilfe der Signale  $m_2$  und  $e_2$  werden die Teilwörter  $w$  und  $v_1$  auf gleiche Länge getestet.
- 4) Mit derselben Technik wie in 2) wird geprüft, ob im Teilwort  $w$  genau  $n$  Zeichen aus  $\{a,b\}$  sind. Das a-Signal wird dabei an jedem Zeichen aus  $\{a,b\}$  um einen Zeittakt verzögert, die angetroffenen a's und b's werden markiert.

In der zweiten Phase wird die Bedingung  $w = v^R$  überprüft. Dazu wird von jedem markierten a und b im Teilwort  $w$  ein  $a_1$ - bzw.  $b_1$ -Signal mit Geschwindigkeit  $\frac{1}{2}$  nach links geschickt. Die  $a_1$ - und  $b_1$ -Signale werden 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  $m_1$  wieder in einer Zelle einholt, die ein a bzw. b enthält, so folgt  $w = v^R$ .

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

Die Anzahl der benötigten Zustandsänderungen ist im wesentlichen bestimmt durch die Anzahl der  $a_1$ - und  $b_1$ -Signale. Für die Erkennung eines Eingabewortes  $w_Q = vXXwYYQ_{(k)}(\$^y)ZZ\$^y\#^t$  aus  $L_{RQ_{(k)}}$  werden daher in keiner Zelle mehr als

$$O(y) = O(|w_Q| \binom{1}{2^k})$$

Zustandsänderungen benötigt. Im Falle eines Wortes  $w_D$  aus  $L_{RD_{(k)}}$  ergeben sich maximal  $O(\ln_{(k)} |w_D|)$  Zustandsänderungen.

◇

**Lemma 3.10 a)** Es sei  $k \in \mathbb{N}$  und  $\epsilon, s \in \mathbb{R}$  mit  $\epsilon > 0$ . Dann gibt es keinen  $s n^{\binom{1}{2^k} - \epsilon}$ -max-änderungsbeschränkten ZA, der  $L_{RQ_{(k)}}$  erkennt.

**b)** Es sei  $k \in \mathbb{N}$  und  $s \in \mathbb{R}$ . Dann gibt es keinen  $s \ln_{(k+1)} n$ -max-änderungsbeschränkten ZA, der  $L_{RD_{(k)}}$  erkennt.

Beweis: Der Beweis läuft analog zum Beweis des Lemmas 3.8 ab. Deshalb werden hier nur die entsprechenden Abschätzungen für die Anzahl der verschiedenen Verhaltensfolgeausschnitte durchgeführt.

a) Ein  $sn^{\left(\frac{1}{2^k} - \epsilon\right)}$ -max-änderungsbeschränkter ZA ist nach Lemma 3.1  $snn^{\left(\frac{1}{2^k} - \epsilon\right)}$ -zeitbeschränkt. Die Maximalzahl der Verhaltensfolgeausschnitte  $\langle c_{\sigma}^W(v, v+1) \rangle$  ist nach Lemma 3.4 b) mit  $\rho = sn^{\left(\frac{1}{2^k} - \epsilon\right)}$  und  $\tau = snn^{\left(\frac{1}{2^k} - \epsilon\right)}$  beschränkt durch

$$\begin{aligned} & \left( snn^{\left(\frac{1}{2^k} - \epsilon\right)} \right)^{2sn^{\left(\frac{1}{2^k} - \epsilon\right)}} |A|^{2\left( sn^{\left(\frac{1}{2^k} - \epsilon\right)} + 1 \right)} \\ & < (sn^2)^{2sn^{\left(\frac{1}{2^k} - \epsilon\right)}} (|A|^2)^{2sn^{\left(\frac{1}{2^k} - \epsilon\right)}} \quad \text{für } n \text{ mit } sn^{\left(\frac{1}{2^k} - \epsilon\right)} \geq 1 \\ & \leq (n^5)^{2sn^{\left(\frac{1}{2^k} - \epsilon\right)}} \quad \text{für } n \geq \max\{|A|, s\} \\ & = n^{10sn^{\left(\frac{1}{2^k} - \epsilon\right)}} \\ & \leq \left( \frac{n}{8} \right)^{\left( \frac{1}{2} \left( \frac{n}{8} \right)^{\frac{1}{2^k}} \right)} \quad \text{für genügend große } n. \end{aligned}$$

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

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

solche Wörter  $v$ , dies ist folglich auch die Anzahl der Wörter der Länge  $n$  in  $L_{RQ(k)}$ . Für große  $n$  gilt nach der Definition von  $L_{RQ(k)}$   $n < 8v^{2^k}$  und damit  $\left( \frac{n}{8} \right)^{\frac{1}{2^k}} < v$ . Daraus folgt:

$$\begin{aligned} & \left( \frac{n}{8} \right)^{\left( \frac{1}{2} \left( \frac{n}{8} \right)^{\frac{1}{2^k}} \right)} < (v^{2^k})^{\frac{v}{2}} \\ & < \binom{v^{2^k}}{v} < \binom{v^{2^k}}{v} 2^v \quad \text{nach Lemma 3.5.} \end{aligned}$$

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

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

$$\begin{aligned} & \left( s \ln_{(k+1)} n \right)^{2s \ln_{(k+1)} n} |A|^{2(s \ln_{(k+1)} n + 1)} \\ < & \left( s n^2 \right)^{2s \ln_{(k+1)} n} \left( |A|^2 \right)^{2s \ln_{(k+1)} n} \quad \text{für } n \text{ mit } s \ln_{(k+1)} n > 1 \\ \leq & \left( n^5 \right)^{2s \ln_{(k+1)} n} \quad \text{für } n \geq \max\{|A|, s\} \\ = & n^{10s \ln_{(k+1)} n} \\ \leq & \left( \frac{n}{8} \right)^{\frac{1}{2} \ln_{(k)} \left( \frac{1}{8} n \right)} \quad \text{für große } n, \text{ da } \frac{\ln_{(k+1)} n}{\ln_{(k)} n} \text{ dann gegen null geht.} \end{aligned}$$

Zu jedem Wort  $v$  der Länge  $2_{(k)}^v$  mit  $v$  Zeichen aus  $\{a, b\}$  gibt es genau ein Wort  $vXXv^RYYD_k(\$^v)ZZ\$^{v\#t}$  mit Länge  $n$  in  $L_{RD(k)}$ . Es gibt genau

$$\binom{2_{(k)}^v}{v} 2^v$$

solche Wörter  $v$ , dies ist folglich auch die Anzahl der Wörter der Länge  $n$  in  $L_{RD(k)}$ . Für große  $n$  gilt nach der Definition von  $L_{RD(k)}$   $n < 8 \cdot 2_{(k)}^v$  und damit  $\ln_{(k)} \left( \frac{n}{8} \right) < v$ . Daraus folgt

$$\begin{aligned} \left( \frac{n}{8} \right)^{\frac{1}{2} \ln_{(k)} \left( \frac{1}{8} n \right)} & < \left( 2_{(k)}^v \right)^{\frac{v}{2}} \\ < \binom{2_{(k)}^v}{v} & < \left( \frac{2_{(k)}^v}{v} \right) 2^v \quad \text{nach Lemma 3.5.} \end{aligned}$$

Also kann  $L_{RD(k)}$  von keinem  $s \ln_{(k+1)} n$ -max-änderungsbeschränkten ZA erkannt werden. ◇

Aus Lemma 3.10 folgt insbesondere, daß für  $k \in \mathbb{N}$   $L_{RQ(k)}$  von keinem  $O(\ln n)$ - und  $L_{RD(k)}$  von keinem  $O(1)$ -änderungsbeschränkten ZA erkannt werden kann. Damit ergibt sich unmittelbar aus Lemma 3.9 und 3.10 die folgende Verfeinerung der Hierarchie von Satz 3.1.

**Satz 3.2** Für  $Z \in \{ \check{A}ZA, \check{A}ZA_R, \check{A}UZA, \check{A}UZA_R \}$  gilt

$$\begin{aligned} Z(O(n)) \supset Z(O(n^{\frac{1}{2}})) \supset Z(O(n^{\frac{1}{3}})) \supset Z(O(n^{\frac{1}{4}})) \supset \dots \\ \dots \supset Z(O(\ln n)) \supset Z(O(\ln_{(2)} n)) \supset Z(O(\ln_{(3)} n)) \supset \dots \supset Z(O(1)). \end{aligned}$$

### 3.4 Zustandsänderungsbeschränkte ZA und die Relation von Nerode

Ein wichtiges Kriterium für reguläre Sprachen liefert die in der nächsten Definition eingeführte Relation  $\sim$ , die sogenannte Äquivalenzrelation von Nerode : Eine Sprache  $L \subseteq V^+$  ist genau dann von einem endlichen Automaten erkennbar, wenn es in  $V^+$  nur endlich viele Äquivalenzklassen bezüglich  $\sim \pmod L$  gibt.

Betrachtet man eine etwas modifizierte Relation  $\sim_k$ , so erhält man ein notwendiges Kriterium für die Realzeiterkennbarkeit von Sprachen durch Iterative Arrays [Co69] und Iterative Bäume [CuY84]. In diesem Abschnitt wird nun untersucht, ob es vergleichbare Kriterien für änderungsbeschränkte Zellularautomaten gibt.

**Definition 3.3** Es sei  $L \subseteq V^*$ ;  $x, y \in V^*$ ;  $k \in \mathbb{N}$ . Dann werden die Relationen  $\sim_k \pmod L$  und  $\sim \pmod L$  auf  $V^*$  definiert durch

$$a) x \sim_k y \pmod L : \Leftrightarrow \forall w \in \overline{V^k} (xw \in L \Leftrightarrow yw \in L)$$

$$b) x \sim y \pmod L : \Leftrightarrow \forall w \in V^* (xw \in L \Leftrightarrow yw \in L).$$

Bei der Betrachtung von  $n$ -dimensionalen iterativen Arrays erhält man für eine Realzeitsprache  $L$  eine obere Schranke für die Anzahl der Äquivalenzklassen bezüglich  $\sim_k \pmod L$  :

$$\text{Äq}(V^*, \sim_k \pmod L) \leq |A|^{((2k+1)^n)} \quad [\text{Co69}]$$

Für eine Realzeitsprache  $L$  eines  $n$ -ären iterativen Baumes gilt entsprechend :

$$\text{Äq}(V^*, \sim_k \pmod L) \leq (|A|^2)^{(n^k)} \quad [\text{CuY84}]$$

Dabei ist  $A$  die Zustandsmenge des iterativen Automaten

Eine solche Einschränkung der Realzeitsprachen ist für Zellularautomaten im allgemeinen nicht möglich, denn schon unidirektionale ZA erkennen im Gegensatz zu iterativen Automaten alle linearen Sprachen in Realzeit. Die linearen Sprachen lassen sich aber bezüglich der Anzahl der Äquivalenzklassen von  $\sim_k$  nur durch das absolute Maximum der möglichen Äquivalenzklassen begrenzen : Die Anzahl der Äquivalenzklassen ist für alle Sprachen über einem Alphabet  $V$  beschränkt durch die Anzahl der Teilmengen von  $\overline{V^k}$ :

$$\forall L \subseteq V^* \left( \text{Äq}(V^*, \sim_k \pmod L) \leq 2^{|\overline{V^k}|} \right)$$

Im Fall der zustandsänderungsbeschränkten unidirektionalen ZA erhält man jedoch vergleichbare Kriterien für eine der Äquivalenzrelation von Nerode sehr ähnliche Relation.

Definition 3.4 Es seien  $L \subseteq V^*$ ;  $x, y \in V^*$ ;  $k \in \mathbb{N}$ . Dann wird die Relation  ${}_k \sim \pmod L$  auf  $V^*$  definiert durch

$$x \sim_k y \pmod L \iff \forall w \in \overline{V^k} (wx \in L \iff wy \in L).$$

Bei  ${}_k \sim$  handelt es sich also lediglich um die „umgedrehte“ Relation  $\sim_k$ . Betrachtet man Zellularautomaten mit der akzeptierenden Zelle am rechten Rand der Retina, so gelten die folgenden Betrachtungen entsprechend für die ursprüngliche Relation  $\sim_k$ .

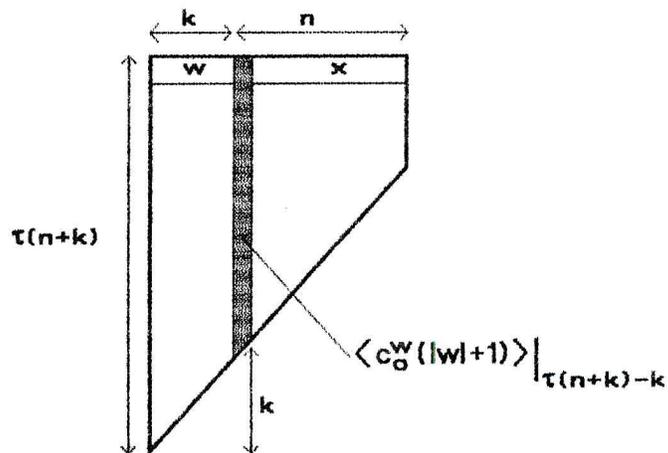
Satz 3.3 Es sei  $L \subseteq V^*$  und  $\rho, \tau: \mathbb{N} \rightarrow \mathbb{N}$  mit  $\rho(n) \leq \tau(n)$  für alle  $n \in \mathbb{N}$ . Wird  $L$  von einem  $\rho$ -max-zustandsänderungsbeschränkten UZA  $\mathfrak{A} = (A, 1, N, F)$  in der Zeit  $\tau(n)$  erkannt, so gilt für alle  $k, n \in \mathbb{N}$

$$\text{Äq}(\overline{V^n}, \sim_k \pmod L) < |A|^{\rho(n+k)+1} (\tau(n+k)-k)^{\rho(n+k)}.$$

Beweis: Bei einer durch ein beliebiges Eingabewort  $wx$  mit  $w \in \overline{V^k}$  und  $x \in \overline{V^n}$  bestimmten Verhaltensfolge  $\langle c_0^{wx} \rangle$  in einem UZA ist der Verhaltensfolgeausschnitt  $\langle c_0^{wx} (|w|+1) \rangle$  allein durch das Teilwort  $x$  bestimmt. Daraus folgt insbesondere

$$\langle c_0^{wx} (|w|+1) \rangle \Big|_{\tau(n+k)-k} = \langle c_0^x (1) \rangle \Big|_{\tau(n+k)-k} \quad (*)$$

Abb. 3.9 Skizze einer Verhaltensfolge  $\langle c_0^{wx} \rangle$  eines UZA



In einem  $\rho(n)$ -änderungsbeschränkten UZA gibt es nach Lemma 3.4 weniger als

$$|A|^{\rho(n+k)+1} (\tau(n+k)-k)^{\rho(n+k)}$$

verschiedene solche Verhaltensfolgeausschnitte. Sind die zu  $x, y \in \overline{V^n}$  gehörenden Verhaltensfolgeausschnitte gleich, gilt also

$$\langle c_0^x (1) \rangle \Big|_{\tau(n+k)-k} = \langle c_0^y (1) \rangle \Big|_{\tau(n+k)-k},$$

dann folgt mit (\*) für alle  $v \in \overline{V}^k$

$$\langle c_0^{v_x}(|v|+1) \rangle_{\tau(n+k)-k} = \langle c_0^{v_y}(|v|+1) \rangle_{\tau(n+k)-k}$$

Wird anschließend Lemma 3.3 mit  $w = vx$  und  $\bar{w} = vy$  angewendet, so erhält man

$$\langle c_0^{v_x}(1) \rangle_{\tau(n+k)} = \langle c_0^{v_y}(1) \rangle_{\tau(n+k)}$$

und somit (nach Definition von  $k \sim$ )  $x \sim_k y \pmod{L}$ .

◇

Die Anzahl der Äquivalenzklassen  $\text{Äq}(\overline{V}^n, k \sim \pmod{L})$  einer ÄUZA-Sprache  $L$  ist also nach diesem Ergebnis durch die Anzahl der verschiedenen Verhaltensfolgeausschnitte  $\langle c_0^x(1) \rangle$  für  $x \in \overline{V}^n$  begrenzt. Eine ähnliche Einschränkung ist für bidirektionale ZA nur bei Sprachen einer speziellen Struktur bezüglich einer gegenüber  $k \sim$  modifizierten Relation  $k \approx$  möglich. Der Beweis dieses Kriteriums findet sich in Abschnitt 3.6.

Die mit Satz 3.3 erreichbaren Schranken sind wesentlich kleiner als die bekannten Schranken für die Anzahl der Äquivalenzklassen von Realzeitsprachen der iterativen Arrays oder Bäume bezüglich der Relation  $\sim_k$ . Es gilt nämlich für alle  $L \subseteq V^*$

$$\text{Äq}(\overline{V}^n, k \sim \pmod{L}) \leq \min(|\overline{V}^n|, 2^{|\overline{V}^k|}),$$

dabei ist häufig  $|\overline{V}^n|$  die entscheidende Größe. Durch die Einbeziehung der Größe  $n$  unterscheidet sich die Schranke aus Satz 3.3 auch qualitativ von denen für iterative Automaten; Abschätzungen der Anzahl der Äquivalenzklassen einer Sprache für iterative Automaten können daher in der Regel nicht auf Satz 3.3 übertragen werden.

### 3.5 Die Abschlußeigenschaften bezüglich Spiegelbild

In diesem Abschnitt werden zunächst die Abschlußeigenschaften der Sprachklassen der änderungsbeschränkten ZA bezüglich Spiegelung untersucht. Anhand der Ergebnisse werden die unterschiedlichen Möglichkeiten von unidirektionalen und bidirektionalen ZA deutlich gemacht. Eine Folge dieser Unterschiede ist, daß sich die von bidirektionalen ZA einer bestimmten Änderungskomplexität erkannten Sprachen wesentlich schwieriger eingrenzen lassen als im unidirektionalen Fall. Insbesondere ist eine Beschränkung der Anzahl der Äquivalenzklassen bezüglich  $\kappa^{\sim}$  oder  $\sim_{\kappa}$  analog zu Satz 3.3 grundsätzlich nicht möglich.

Schreibweise: Im folgenden sei  $\mathfrak{K} := \{n, n^{\frac{1}{2}}, n^{\frac{1}{4}}, \dots, \ln n, \ln_{(2)} n, \ln_{(3)} n, \dots, 1\}$  die Menge der aus 3.2 und 3.3 bekannten Zustandsänderungskomplexitäten.

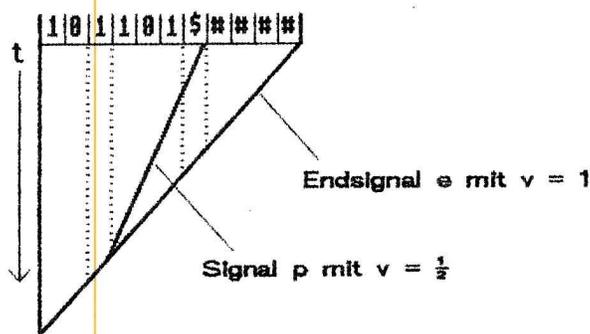
**Satz 3.4** Die Sprachklassen  $\text{ÄUZA}(O(g))$  und  $\text{ÄUZA}_R(O(g))$  mit  $g \in \mathfrak{K}$ ,  $g < n$  sind nicht abgeschlossen bezüglich Spiegelung.

Beweis: Es sei  $L_B := \{u\$^k \mid u \in \{0,1\}^+ \wedge |u| \geq k \wedge u_{|u|-k+1} = 1\}$ .

a) Zunächst wird gezeigt :  $L_B \in \text{ÄUZA}_R(O(1))$

$L_B$  ist durch den folgenden einfachen Algorithmus mit nicht mehr als drei Zustandsänderungen pro Zelle von einem UZA erkennbar.

**Abb. 3.10** Erkennung von  $L_B$  in einem UZA



Funktionsweise: Es sei  $w$  ein Wort der Form  $u\$^k$  mit  $u \in \{0,1\}^+$ . Falls das Wort aus  $L_B$  ist, muß  $u_{|u|-k+1}$ , also das  $k$ -te Zeichen von rechts aus dem Teilwort  $u$ , eine 1 sein.

Wird am Anfang ein Signal  $p$  mit Geschwindigkeit  $\frac{1}{2}$  vom  $\$$ -Zeichen aus gestartet, so wird dieses genau in  $u_{|u|-k+1}$  vom schnelleren Endsignal  $e$  eingeholt und das Zeichen kann auf 0 oder 1 getestet werden. Die Überprüfung der Eingabe auf die Form  $u\$^k, u \in \{0,1\}^+$  ist ein reguläres Problem und kann daher vom Signal  $e$  auf seinem Weg nach links gleich noch miterledigt werden.

b) Noch zu zeigen:  $(L_B)^R \notin \text{ÄUZA}(O(n^{1-\varepsilon}))$

Der Beweis erfolgt mit Hilfe von Satz 3.3:

Eine Abschätzung der Zahl der Äquivalenzklassen bezüglich  $\kappa^{\sim}$  für die Sprache  $(L_B)^R = \{ \#^k \$ u \mid u \in \{0,1\}^+ \wedge |u| \geq k \wedge u_k = 1 \}$  erhält man durch die folgende Überlegung:

Sind  $u$  und  $\bar{u}$  verschiedene Wörter der Länge  $k$  aus  $\{0,1\}^+$ , dann gibt es ein  $x \leq k$  mit  $u_x \neq \bar{u}_x$ . Gilt ohne Einschränkung  $u_x = 1 \wedge \bar{u}_x = 0$ , so folgt

$$\#^x \$ u \in (L_B)^R \wedge \#^x \$ \bar{u} \notin (L_B)^R.$$

Damit sind nach Definition 3.3  $\$ u$  und  $\$ \bar{u}$  nicht in der gleichen Äquivalenzklasse bezüglich  $\kappa^{\sim}(\text{mod}(L_B)^R)$ . Da es insgesamt  $2^k$  verschiedene Wörter der Länge  $k$  aus  $\{0,1\}^+$  gibt, folgt

$$\text{Äq}(\sqrt[k+1]{}, \kappa^{\sim}(\text{mod}(L_B)^R)) \geq 2^k.$$

Eine Sprache  $L$  aus  $\text{ÄUZA}(O(n^{1-\varepsilon}))$  ist nach Lemma 3.1  $O(n^{2-\varepsilon})$ -zeitbeschränkt. Es folgt aus Satz 3.3 mit einem  $s \in \mathbb{R}$  und mit  $m = k+1 \in \mathbb{N}$

$$\begin{aligned} \text{Äq}(\sqrt[m]{}, \kappa^{\sim}(\text{mod} L)) &< |A|^{s(m+k)^{1-\varepsilon}+1} (s(m+k)^{2-\varepsilon}-k)^{s(m+k)^{1-\varepsilon}} \\ &= |A|^{s(2k+1)^{1-\varepsilon}+1} (s(2k+1)^{2-\varepsilon}-k)^{s(2k+1)^{1-\varepsilon}} \quad \text{denn es gilt } m+k=2k+1 \\ &< (k^{\varepsilon})^3 s k^{1-\varepsilon} \quad \text{für } k \text{ mit } k \geq \max\{|A|, s\} \text{ und } k^2 \geq 2k+1 \\ &= 2^{18} s k^{1-\varepsilon} \cdot |dk| < 2^k \quad \text{für große } k, \text{ da dann } k^{-\varepsilon} \cdot |dk| \text{ gegen null geht.} \end{aligned}$$

Also kann  $(L_B)^R$  von keinem  $O(n^{1-\varepsilon})$ -max-änderungsbeschränkten UZA erkannt werden. ◇

$\text{ÄUZA}_R(O(n))$  ist identisch mit der Familie der von unidirektionalen ZA mit Realzeitbeschränkung erkannten Sprachen und damit abgeschlossen bezüglich Spiegelung. Ein Beweis befindet sich beispielsweise in [ChC85]. Es ist ein bekanntes offenes Problem in der Theorie der Zellularautomaten, ob  $\text{ÄZA}_R(O(n))$  und damit die Familie der in Realzeit von ZA erkannten Sprachen bezüglich Spiegelung abgeschlossen ist. Aber auch für die Klasse  $\text{ÄUZA}(O(n))$  ist kein diesbezügliches Ergebnis bekannt.

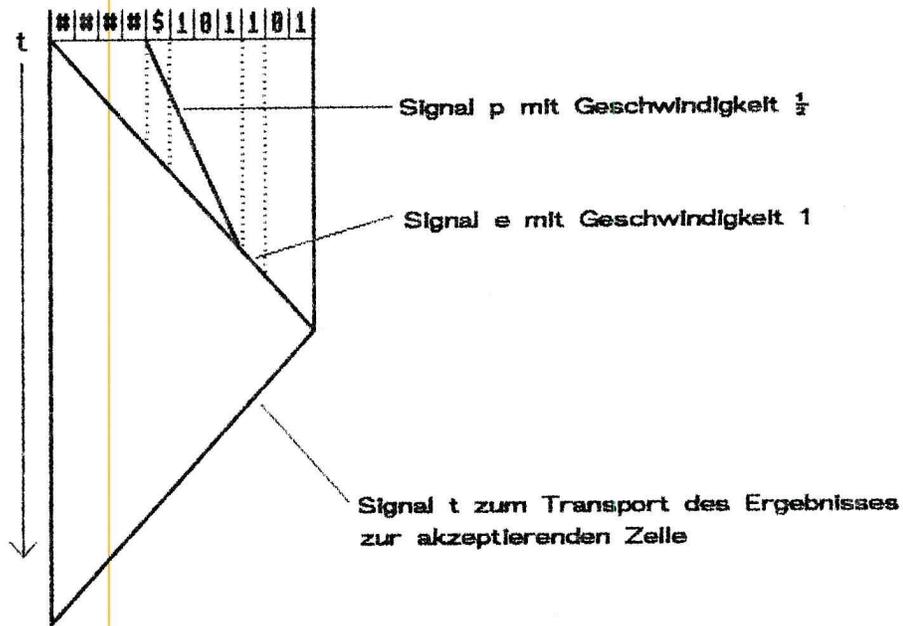
**Satz 3.5 Die Sprachfamilien  $\text{ÄZA}(O(g))$  sind für alle  $g \in \mathfrak{S}$  abgeschlossen bezüglich Spiegelung.**

Beweis: Ein Algorithmus zur Erkennung einer Sprache  $L$  kann durch einfaches Spiegeln in ein Verfahren für  $(L)^R$  transformiert werden. Das dann am rechten Rand der Retina erhaltene Ergebnis kann durch ein Signal  $t$  zur akzeptierenden Zelle am linken Rand der Retina transportiert werden.

Die Größenordnung der Zustandsänderungszahl wird dadurch nicht verändert.

Für den Fall  $L = L_B$  erhält man beispielsweise das folgende Verfahren für  $(L_B)^R$ :

**Abb. 3.11** Erkennung von  $(L_B)^R$  mit  $O(1)$  Zustandsänderungen pro Zelle



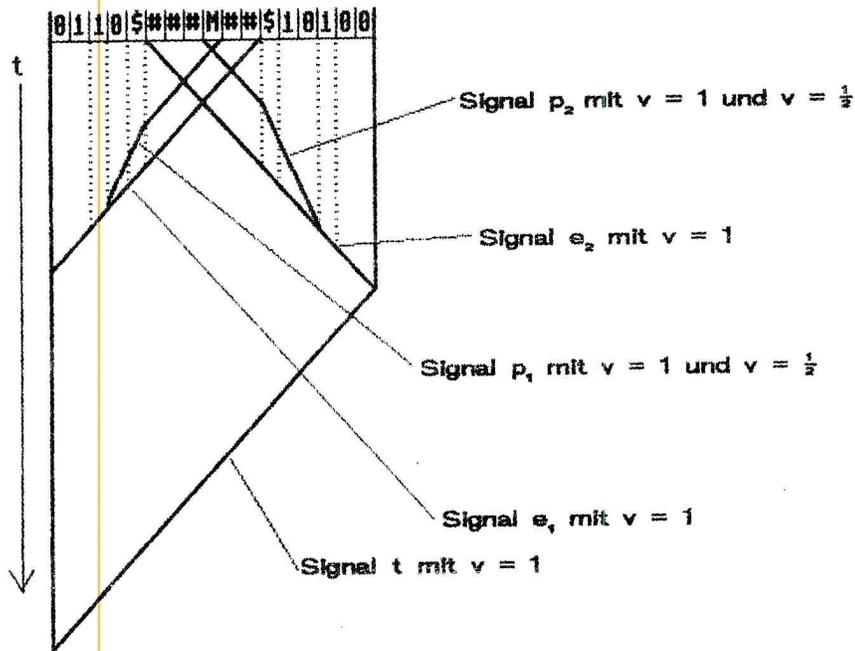
Nach den Betrachtungen im Beweis von Satz 3.4 läßt sich die Anzahl der Äquivalenzklassen in  $\overline{X^n}$  bezüglich  $\sim_{(L_B)^R}$  nicht sinnvoll gegenüber dem absoluten Maximum  $\min(|\overline{X^n}|, 2^{|\overline{X^n}|}) = |\overline{X^n}|$  beschränken, so daß zur Erkennung von  $(L_B)^R$  in einem UZA  $O(n)$  Zustandsänderungen pro Zelle benötigt werden.

Andererseits kann  $(L_B)^R$  mit dem oben skizzierten Verfahren von bidirektionalen ZA mit  $O(1)$  Zustandsänderungen erkannt werden. Damit ist klar, daß es für die Sprachen der auf Realzeit beschränkten bidirektionalen ZA kein Kriterium analog zu Satz 3.3 geben kann.

**Beispiel 3.4**  $L_{BB} := \{u\$ \#^k M \#^n \$v \mid u, v \in \{0,1\}^+ \wedge |u| \geq n \wedge |v| \geq k \wedge u_{|u|-n+1} = v_k = 1\}$  ist von einem  $O(1)$ -änderungsbeschränkten ZA erkennbar. Aufgrund der großen Anzahl von Äquivalenzklassen bezüglich  $\sim$  ist  $L_{BB}$  von keinem UZA mit weniger als  $O(n)$  Zustandsänderungen pro Zelle erkennbar. Da  $L_{BB} = (L_{BB})^R$  gilt, wird auch das Spiegelbild nicht mit weniger Zustandsänderungen von UZA erkannt. Es bringt also in diesem Fall nichts, den UZA zu spiegeln und die akzeptierende Zelle am rechten Rand anzunehmen. Zur Erkennung von  $L_{BB}$  mit  $O(1)$  Zustandsänderungen wird echte bidirektionale Kommunikation benötigt.

Funktionsweise: Liegt ein wohlgeformtes Eingabewort  $u\#^kM\#^n v$  vor, so werden die Bedingungen  $u_{|u|-n+1} = 1$  und  $v_k = 1$  analog zum Verfahren für  $L_B$  mit Hilfe der Signale  $e_1$  und  $p_1$  bzw.  $e_2$  und  $p_2$  überprüft. Die Signale  $p_1$  und  $p_2$  werden im Gegensatz zum Verfahren für  $L_B$  zuerst mit der Geschwindigkeit 1 gestartet, um die eingeschobenen Teilwörter zu überbrücken. Beim Erreichen des Teilwortes  $u$  bzw.  $v$  werden sie dann auf die Geschwindigkeit  $\frac{1}{2}$  abgebremst, so dass sie im zu prüfenden Zeichen vom  $e_1$ - bzw.  $e_2$ -Signal eingeholt werden. Das Signal  $t$  transportiert das am rechten Rand erhaltene Teilergebnis nach links und überprüft die Eingabe auf Wohlgeformtheit.

Abb. 3.12 Erkennung von  $L_{BB}$  in einem ZA mit  $O(1)$  Zustandsänderungen



Satz 3.6 Es sei  $g \in \mathbb{N}$ ,  $g \neq n$ . Dann gelten

- $\check{A}UZA(O(g)) \subset \check{A}ZA(O(g))$ .
- $1 \neq g \Rightarrow \check{A}UZA(O(g)) \not\subset \check{A}ZA(O(1)) \wedge \check{A}UZA_R(O(g)) \not\subset \check{A}ZA(O(1))$

Beweis: Die Sprache  $(L_B)^R$  ist nach Satz 3.5 in  $\check{A}ZA(O(1))$  und damit auch in  $\check{A}ZA(O(g))$ , aber nach Satz 3.4 nicht in  $\check{A}UZA(O(g))$ . Andererseits ist aber nach Lemma 3.10 keine der Sprachen  $L_{RD(n)}$  und  $L_{RQ(n)}$  in  $\check{A}ZA(O(1))$ .  $\diamond$

Es gibt also nach Satz 3.6 Sprachen, bei denen die Verwendung eines bidirektionalen ZA zu einer wesentlichen Verringerung der Anzahl der benötigten Zustandsänderungen gegenüber der Erkennung in einem UZA führt. Ein Beispiel dafür ist die Sprache  $(L_B)^R$ , die in einem ZA mit erheblich weniger „Aufwand“ erkannt werden kann als in einem UZA. In vielen anderen Fällen aber, beispielsweise bei den Sprachen  $L_{RQ(n)}$  und  $L_{RD(n)}$ , kann mit einem bidirektionalen ZA überhaupt keine Veränderung der Zustandsänderungskomplexität gegenüber dem unidirektionalen Fall erreicht werden.

### 3.6 Ein Kriterium für bidirektionale Realzeit-Zellularautomaten

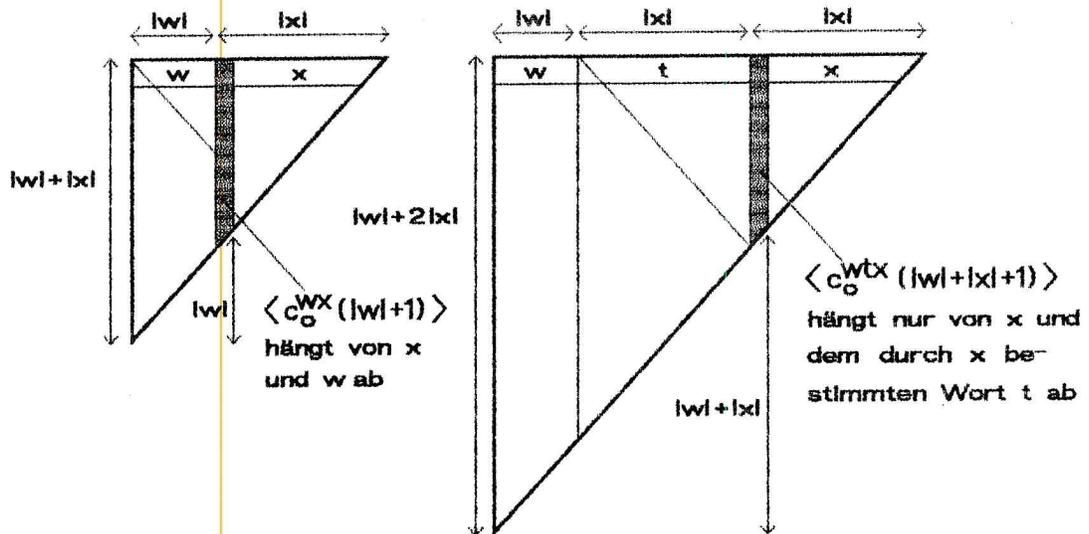
Bisher wurde die Frage offengelassen, ob die Klassen  $\check{A}ZA_R(O(g))$  der auf Realzeit beschränkten ZA abgeschlossen bezüglich Spiegelung sind. Das im Beweis von Satz 3.5 beschriebene Verfahren zur Erkennung des Spiegelbildes einer Sprache benötigt mehr als Realzeit und ist daher nicht anwendbar.

Betrachtet man beispielsweise die Sprache  $L_B$ , so wird man bald zu der Vermutung kommen, daß das Spiegelbild  $(L_B)^R$  von keinem  $O(1)$ -änderungsbeschränkten ZA in Realzeit erkannt werden kann und daß insgesamt die Klassen  $\check{A}ZA_R(O(g))$ ,  $g \in \mathfrak{R}$  für  $g(n) < n$  nicht abgeschlossen bezüglich Spiegelung sind. Tatsächlich kann man auch beweisen, daß zur Erkennung von  $(L_B)^R$  in Realzeit  $O(n)$  Zustandsänderungen benötigt werden.

Um die Nichtabgeschlossenheit der Klassen  $\check{A}ZA_R(O(g))$  bezüglich Spiegelung zu beweisen, wird hier jedoch ein anderer Weg gegangen. Beschränkt man sich auf Sprachen einer speziellen Struktur, so ist es möglich, ein Kriterium ähnlich zu Satz 3.3 für bidirektionale Realzeit-ZA zu bekommen, allerdings mit einer etwas anderen Relation  $\approx$ , die in Definition 3.5 eingeführt wird.

Eine direkte Übertragung von Satz 3.3 auf bidirektionale ZA scheitert daran, daß der im Beweis des Satzes betrachtete Verhaltensfolgeausschnitt  $\langle c_o^{wx}(|w|+1) \rangle$  in einem ZA mit bidirektionalem Datenfluß nicht nur vom Teilwort  $x$ , sondern auch von  $w$  abhängt. Fügt man aber zwischen  $w$  und  $x$  ein zusätzliches Teilwort  $t = \&^{|x|}$  aus Sonderzeichen  $\&$  ein, so hängt der Verhaltensfolgeausschnitt  $\langle c_o^{wtx}(|w|+|x|+1) \rangle$  in einem Realzeit-ZA nur von  $x$  ab.

Abb. 3.13 Verhaltensfolgen in einem bidirektionalen Realzeit-ZA



Ausgehend von dieser Idee kann man ein zu Satz 3.3 analoges Kriterium für Sprachen erhalten, die durch Einfügen einer geeigneten Zahl von  $\&$ -Zeichen aus einer anderen Sprache erzeugt werden. Mit Hilfe dieses Ergebnisses wird dann am Beispiel einer Sprache  $L_{B\&}$ , die aus  $L_B$  durch Einfügen von  $\&$ -Zeichen entsteht, die Nichtabgeschlossenheit der Klassen  $\check{A}ZA_R(O(g))$ ,  $g \in \mathfrak{R}$  und  $g(n) < n$  gezeigt.

**Definition 3.5** Es seien  $L \subseteq (V \cup \{\&\})^*$ ;  $x, y \in V^*$ ;  $k \in \mathbb{N}$ . Dann wird die Relation  $x \stackrel{k}{\approx} y \pmod{L}$  auf  $V^*$  definiert durch

$$x \stackrel{k}{\approx} y \pmod{L} \iff \forall w \in \overline{V^k} (w \&^{|x|} x \in L \iff w \&^{|y|} y \in L).$$

Die Betrachtung einer solchen Relation ist natürlich nur sinnvoll für Sprachen, die Wörter der Form  $v \&^n w$  mit  $|w| = n$  enthalten bzw. nur aus solchen Wörtern bestehen. Für alle anderen Sprachen erhält man mit dem folgenden Satz keine brauchbaren Ergebnisse.

**Satz 3.7** Es sei  $L \subseteq V^*$  und  $\rho(n) \leq n$  für alle  $n \in \mathbb{N}$ . Wird  $L$  von einem  $\rho$ -änderungsbeschränkten ZA  $\mathfrak{A} = (A, 1, N, F)$  in Realzeit erkannt, so gilt für alle  $k, n \in \mathbb{N}$

$$|\mathfrak{A}_q(V^n, \stackrel{k}{\approx} \pmod{L})| < |A|^{\rho(2n+k)+1} n^{\rho(2n+k)}$$

Beweis: Bei einer durch ein beliebiges Eingabewort  $wtx$  mit  $w \in \overline{V^k}$ ,  $x \in V^n$  und  $t = \&^{|x|}$  bestimmten Verhaltensfolge  $\langle c_0^{wt^k x} \rangle$  in einem ZA ist der Verhaltensfolgeausschnitt  $\langle c_0^{wt^k x}(|w|+n+1) \rangle$  allein durch die Länge des Teilwortes  $x$  bestimmt und damit unabhängig von  $w$ . Daraus folgt insbesondere

$$\langle c_0^{wt^k x}(|w|+n+1) \rangle \Big|_n = \langle c_0^{t^k x}(n+1) \rangle \Big|_n \quad (*).$$

In einem  $\rho(n)$ -änderungsbeschränkten ZA gibt es nach Lemma 3.4 a) weniger als

$$|A|^{\rho(2n+k)+1} n^{\rho(2n+k)}$$

verschiedene solche Verhaltensfolgeausschnitte. Gilt für  $x, y \in V^n$  mit  $t = \&^n$

$$\langle c_0^{t^k x}(|x|+1) \rangle \Big|_n = \langle c_0^{t^k y}(|y|+1) \rangle \Big|_n,$$

dann folgt mit (\*) für alle  $v \in \overline{V^k}$

$$\langle c_0^{vt^k x}(|v|+n+1) \rangle \Big|_n = \langle c_0^{vt^k y}(|v|+n+1) \rangle \Big|_n.$$

und damit

$$\langle c_0^{vt^k x}(1) \rangle \Big|_{2n+k} = \langle c_0^{vt^k y}(1) \rangle \Big|_{2n+k}$$

Daraus folgt aufgrund der Realzeitbeschränkung des ZA:  $x \stackrel{k}{\approx} y \pmod{L}$ .  $\diamond$

Im Gegensatz zu Satz 3.3 wird im obigen Satz 3.7 die Anzahl der Äquivalenzklassen in  $V^n$  und nicht in  $\overline{V^n}$  betrachtet. Dies liegt daran, daß im Beweis von Satz 3.7 die Teilwörter  $x$  und  $y$  die gleiche Länge  $n$  haben müssen, damit die zugehörigen Teilwörter  $t = \&^n$  identisch sind. Für die Abschätzung der Äquivalenzklassen macht dies aber in der Regel keinen Unterschied, da auch bei Anwendung von Satz 3.3 der Einfachheit halber meistens nur die Wörter der Länge  $n$  betrachtet werden, so zum Beispiel beim Beweis von Satz 3.4.

Die Anwendung von Satz 3.7 wird in den meisten Fällen folgendermaßen aussehen: Hat man mit Hilfe des Kriteriums für UZA in Satz 3.3 bewiesen, daß zur Erkennung einer Sprache  $L$  mindestens  $O(g)$  Zustandsänderungen benötigt werden, so kann man durch Einfügen von  $\&$ -Zeichen an geeigneten Stellen eine neue Sprache  $L_g$  konstruieren. Mit Satz 3.7 kann man dann zeigen, daß für  $L_g$  in einem Realzeit-ZA ebenfalls  $O(g)$  Zustandsänderungen benötigt werden.

**Beispiel 3.5** Im Beweis von Satz 3.4 wurde gezeigt, daß die Sprache  $(L_B)^R$  von keinem  $O(n^{1-\epsilon})$ -max-änderungsbeschränkten UZA erkannt werden kann. Zur Abschätzung der Äquivalenzklassen bezüglich  $\approx_k$  wurden dabei die Wörter der Sprache, die nach Definition die Form  $\#^m \$ u$  mit  $u \in \{0,1\}^+$  haben, zwischen den Teilwörtern  $\#^m$  und  $\$$  aufgetrennt.

Fügt man an dieser Stelle eine ( im Sinne von Satz 3.7 ) geeignete Anzahl von &-Zeichen ein, so erhält man die Sprachen

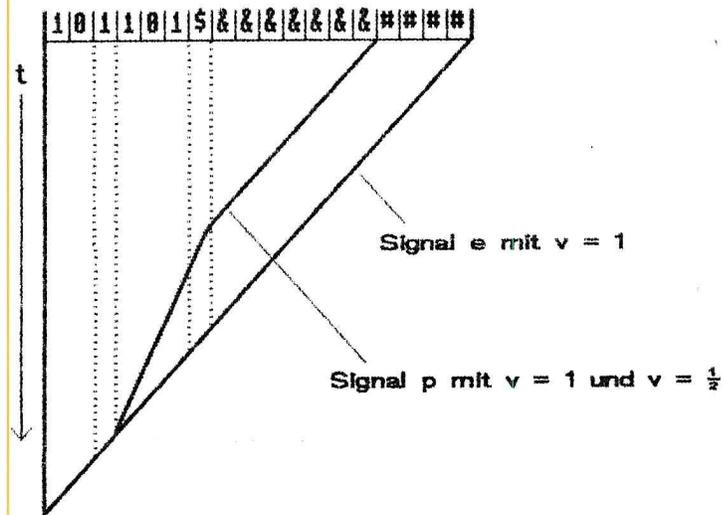
$$L_{B\&} := \{ u \$ \&^n \#^k \mid u \in \{0,1\}^+ \wedge |u| \geq k \wedge n = |u| + 1 \wedge u_{|u|-k-1} = 1 \}$$

und

$$(L_{B\&})^R = \{ \#^k \&^n \$ u \mid u \in \{0,1\}^+ \wedge |u| \geq k \wedge n = |u| + 1 \wedge u_k = 1 \}.$$

Die Erkennung von  $L_{B\&}$  in einem Realzeit-ZA ist nicht aufwendiger als die von  $L_B$  und funktioniert nach folgender Skizze:

Abb. 3.14 Erkennung von  $L_{B\&}$  am Beispiel der Eingabe 101101\$&&&&&&&&&###



Funktionsweise: Der Algorithmus verläuft weitgehend identisch zu dem von  $L_B$ , das Signal p wird jedoch mit Geschwindigkeit 1 gestartet und erst nach dem Passieren der eingefügten &-Zeichen auf die Geschwindigkeit  $\frac{1}{2}$  abgebremst.

Im folgenden wird Satz 3.7 auf die Sprache  $(L_{B\&})^R$  angewendet. Der Beweis verläuft ähnlich wie der Beweis für  $(L_B)^R$  mit Hilfe von Satz 3.3.

Sind  $u$  und  $\bar{u}$  verschiedene Wörter der Länge  $k$  aus  $\{0,1\}^+$ , dann gibt es ein  $x \leq k$  mit  $u_x \neq \bar{u}_x$ . Gilt ohne Einschränkung  $u_x = 0$ , so folgt

$$\#^x \&^{k+1} \$ u \in (L_{B\&})^R \wedge \#^x \&^{k+1} \$ \bar{u} \notin (L_{B\&})^R$$

Damit sind nach Definition 3.5  $\$ u$  und  $\$ \bar{u}$  nicht in der gleichen Äquivalenzklasse bezüglich  $\approx_k \pmod{(L_{B\&})^R}$ . Da es insgesamt  $2^k$  verschiedene Wörter der Länge  $k$  aus  $\{0,1\}^+$  gibt, folgt

$$\text{Äq}(\sqrt[k+1]{\approx_k \pmod{(L_{B\&})^R}}) \geq 2^k$$

Ist eine Sprache  $L$  von einem  $s n^{1-\varepsilon}$ -änderungsbeschränkten ZA mit  $s \in \mathbb{R}$  in Realzeit erkennbar, so folgt nach Satz 3.7 mit  $m = k+1 \in \mathbb{N}$  und  $n = 2m+k$

$$\begin{aligned} \dot{A}q(V^m, k \approx (\text{mod } L)) &< |A|^{s(2m+k)^{1-\varepsilon}+1} m^{s(2m+k)^{1-\varepsilon}} \\ &= |A|^{s(3k+2)^{1-\varepsilon}+1} m^{s(3k+2)^{1-\varepsilon}} \quad \text{wegen } 3k+2 = 2m+k \\ &< (|A|^2)^{4sk^{1-\varepsilon}} m^{4sk^{1-\varepsilon}} \quad \text{für } k \text{ mit } 4sk^{1-\varepsilon} \geq 1 \\ &< (k^4)^{4sk^{1-\varepsilon}} \quad \text{für } k \text{ mit } k \geq \max\{|A|, s\} \text{ und } k^2 > m \\ &= 2^{16} s k^{1-\varepsilon} \cdot |d| k < 2^k \quad \text{für große } k, \text{ da dann } k^{-\varepsilon} \cdot |d| k \text{ gegen null geht.} \end{aligned}$$

Also ist  $(L_{B\&})^R$  von keinem  $O(n^{1-\varepsilon})$ -max-änderungsbeschränkten bidirektionalen Zellularautomaten in Realzeit erkennbar.

Aus Beispiel 3.5 folgen unmittelbar:

**Satz 3.8** Die Sprachfamilien  $\dot{A}ZA_R(O(g))$  sind für alle  $g \in \mathfrak{S}$ ,  $g \neq n$  nicht abgeschlossen bezüglich Spiegelung.

Beweis:  $L_{B\&}$  ist nach Beispiel 3.5 in  $\dot{A}ZA_R(O(1))$ , das Spiegelbild  $(L_{B\&})^R$  ist jedoch in keiner der Klassen  $\dot{A}ZA_R(O(g))$  mit  $g \neq n$ .

◇

**Satz 3.9** Es sei  $g \in \mathfrak{S}$ ,  $g \neq n$ . Dann gelten

- $\dot{A}ZA_R(O(g)) \subset \dot{A}ZA(O(g))$ .
- $1 \neq g \implies \dot{A}ZA_R(O(g)) \not\subset \dot{A}ZA(O(1))$ .

Beweis: Die Sprache  $(L_{B\&})^R$  ist nach Beispiel 3.5 und Satz 3.5 in  $\dot{A}ZA(O(1))$  und damit in  $\dot{A}ZA(O(g))$ , aber nach Beispiel 3.5 nicht in  $\dot{A}ZA_R(O(g))$ . Andererseits ist nach Lemma 3.10 keine der Sprachen  $L_{RD(n)}$  und  $L_{RQ(n)}$  in  $\dot{A}ZA(O(1))$ .

◇

Satz 3.7 wird noch im nächsten Kapitel benutzt, um die Hierarchie der ZA mit  $O(1)$ -Zustandsänderungsbeschränkung zu verfeinern. Es ist auch möglich, den Beweis von Satz 3.8 ohne Verwendung des Kriteriums in Satz 3.7 direkt anhand von  $(L_B)^R$  zu führen. Der Beweis ist dann aber aufwendiger und abhängig von den speziellen Eigenschaften der Sprache.

Ein Beweis mit Hilfe von Satz 3.7 erfolgt dagegen immer nach dem gleichen Schema, es muß für die betrachtete Sprache jeweils nur die Anzahl der Äquivalenzklassen bezüglich  $k \approx$  abgeschätzt werden.