6. Kommunikation in Zellularautomaten

In der Einführung wurde als eines der Motive für die Betrachtung der Zustandsänderungskomplexität die Möglichkeit genannt, damit gleichzeitig ein Maß für die zur Erkennung einer Sprache nötige "Kommunikation" zu erhalten. Unter "Kommunikation" wird hierbei einfach die Menge der in einem Zellularautomaten zwischen verschiedenen Blöcken ausgetauschten Information verstanden.

Anhand des im nächsten Abschnitt folgenden Beispiels läßt sich jedoch demonstrieren, daß diese Erwartung nur bedingt erfüllt wird. Die zwischen verschiedenen Blöcken eines Zellularautomaten transportierte Information ist nicht allein von der Anzahl der Zustandsänderungen, sondern auch von der "Art" der stattfindenden Kommunikation abhängig.

Wie sich auch zeigen wird, ist die Zustandsänderungskomplexität offenbar kein Maß für die zur Erkennung einer Sprache in den in Kapitel 6 definierten Pipeline-verarbeitenden Automaten benötigte Pipeline-Periode bzw. für die Anzahl der Zellen, die zum Erreichen einer Pipeline-Periode von 1 gebraucht werden.

Dazu wird ein einfaches Modell der Kommunikation in Zellularautomaten eingeführt, mit dem einige Eigenheiten zustandsänderungsbeschränkter ZA anschaulich erklärt werden können. Dabei wird in diesem Kapitel weitgehend auf formale Definitionen verzichtet, die verwendeten Begriffe wie zum Beispiel "Kommunikation", "Information" und "Effizienz" sind intuitiv zu verstehen.

6.1 Ein Beispiel

Beispiel 6.1 Es sei $L_{DD} := \{(D_\#(u))^R XX D_\#(u) \mid u \in \{a,b\}^+\}$ mit der Abbildung $D_\#$ aus Definition 3.1. Die Sprache ähnelt in ihrer Struktur etwas der Sprache L_{RD} aus Definition 3.1; sie beruht ebenfalls auf der Palindromsprache L_R aus Beispiel 3.1, die durch Einfügen von #-Zeichen mit Hilfe der Abbildung $D_\#$ "gestreckt" wird.

Dementsprechend wäre eigentlich zu erwarten, daß zur Erkennung von L_{DD} in einem ZA wenigstens $O(\ln n)$ Zustandsänderungen pro Zelle benötigt werden; schließlich müssen in einem Wort $w = (D_\#(u))^R XX D_\#(u)$, $u \in \{a,b\}^+$ der Länge n die $O(\ln n)$ Zeichen aus $\{a,b\}$ im Teilwort $D_\#(u)$ mit $O(\ln n)$ Zeichen im vorderen Teilwort $(D_\#(u))^R$ verglichen werden.

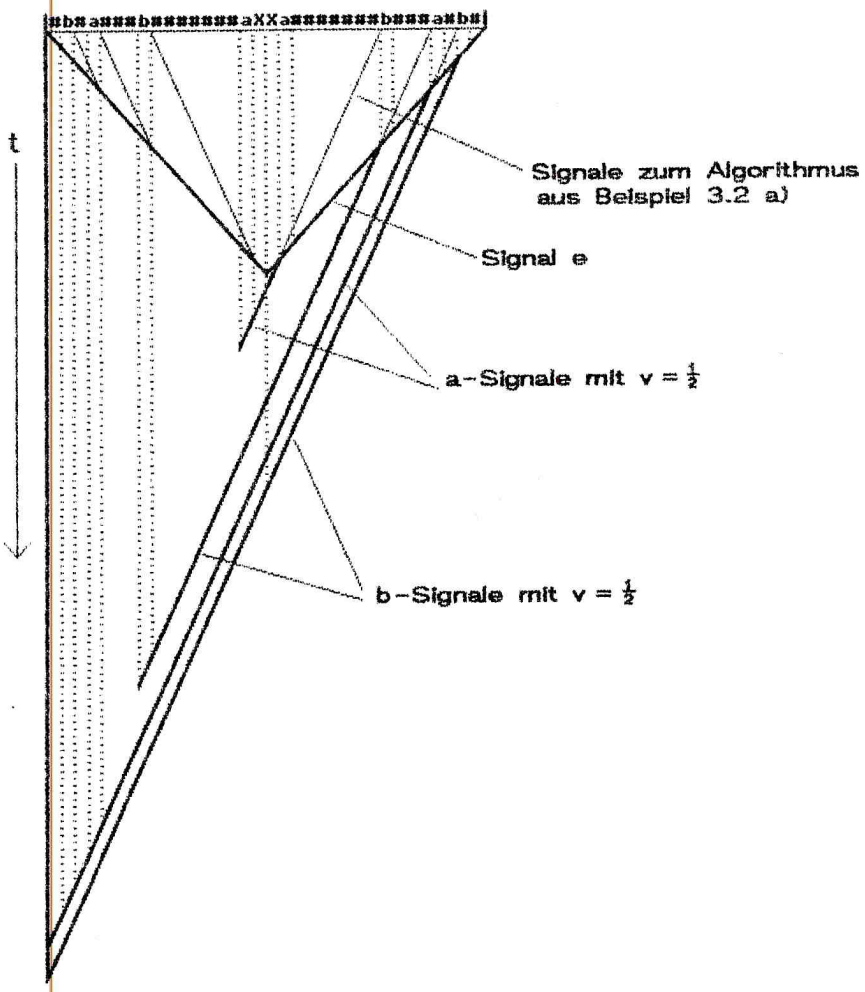
Erkennungsverfahren für L_{DD} mit $O(\ln n)$ Zustandsänderungen lassen sich auch relativ einfach finden. Ein mögliches Verfahren wird in Abbildung 6.1 dargestellt und seine Funktionsweise hier erläutert:

Mit Hilfe des Verfahrens aus Beispiel 3.2 a) wird die Eingabe auf die Form $(D_\#(u_1))^R XX D_\#(u_2)$ geprüft. Der Abschluß dieser Überprüfung wird durch die beiden e-Signale angezeigt.

Von jedem a bzw. b wird beim Eintreffen des von rechts kommenden e-Signals ein a- bzw. b- Signal mit Geschwindigkeit $\frac{1}{2}$ nach links geschickt.

Nach dem Passieren des Teilwortes XX wird jedes a- bzw. b-Signal mit dem nächsten weiter links stehenden und noch nicht markierten Zeichen aus {a,b} verglichen. Anschließend wird das Zeichen markiert, so daß weitere a- und b-Signale darüber hinweggehen und mit weiter links stehenden Zeichen aus {a,b} verglichen werden. Waren alle Vergleiche positiv, so ist das Eingabewort aus der Sprache L_{DD}.

Abb. 6.1 Erkennung von L_{DD} mit $O(\ln n)$ Zustandsänderungen pro Zelle



Versucht man nun, mit der in Lemma 3.6 verwendeten Beweistechnik zu zeigen, daß L_{DD} von keinem ZA mit $O(1)$ Zustandsänderungen pro Zelle erkannt wird, so ergibt sich die folgende Situation:

In L_{DD} gibt es genau 2^k Wörter der Länge $n = 2 \cdot 2^k + 2$. Da in L_{DD} nur Wörter der Längen $2 \cdot 2^\varkappa + 2$ mit $\varkappa \in \mathbb{N}$ existieren, gibt es in der Sprache nur

$$\sum_{\varkappa=1}^{k} 2^\varkappa = 2^{k+1} - 2 = O(n)$$

Wörter der Länge $\leq n$.

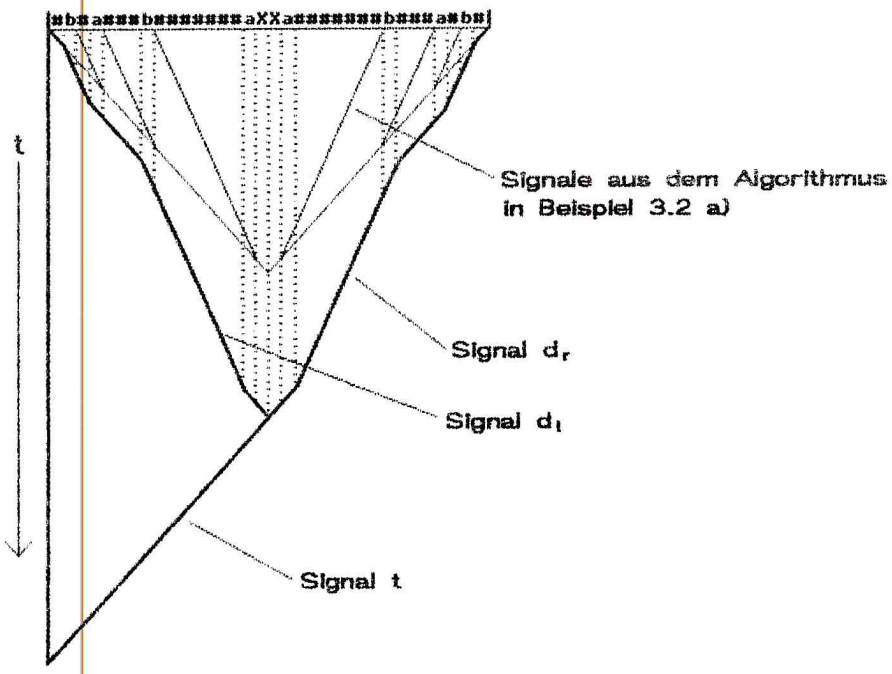
Die Anzahl der Verhaltensfolgeausschnitte in einem ρ-zustandsänderungsbeschränkten ZA mit $\rho \in \mathbb{N}$ ist aber nach Lemma 3.1 und Lemma 3.4 gegeben durch

$$|A|^{\rho+1} \binom{\rho \cdot n}{\rho} = O(n^{\rho}).$$

Der Beweis, daß zur Erkennung von L_{DD} mindestens $O(\ln n)$ Zustandsänderungen pro Zelle benötigt werden, scheitert also mit den in dieser Arbeit verwendeten Methoden. Unter Verwendung eines "geeigneteren" Verfahrens ist es auch tatsächlich möglich, die Sprache L_{DD} mit einer konstanten Anzahl von Zustandsänderungen pro Zelle zu erkennen.

Dieses Verfahren wird mit Hilfe der folgenden Zeichnung erläutert: anschließend wird anhand der Sprache L_{DD} im folgenden Abschnitt ein naives Modell des in zellularen Algorithmen stattfindenden Informationsaustausches vorgestellt. Auch wenn in diesem Modell die Vorgänge in einem Zellularautomaten stark vereinfacht dargestellt werden, so kann es doch das Verständnis der zustandsänderungsbeschränkten Zellularautomaten erleichtern.

Abb. 6.2 Erkennung von L_{DD} mit O(1) Zustandsänderungen pro Zelle am Beispiel der Eingabe $(D_{\#}(abab))^R XX D_{\#}(abab)$ = #b#a###b########aXXa########b###a#b#



Funktionsweise: Mit Hilfe des Verfahrens aus Beispiel 3.2 a) wird die Eingabe auf die Form $(D_{\#}(u_1))^R XX D_{\#}(u_2)$ mit $u_1, u_2 \in \{a,b\}^+$ geprüft.

Gleichzeitig wird vom rechten Rand ein Signal d_r, vom linken Rand ein Signal d_l in die Mitte geschickt. Beide Signale beginnen mit der Geschwindigkeit 1; treffen sie auf ihrem Weg auf ein b, so werden sie auf die

Geschwindigkeit $\frac{1}{2}$ verlangsamt, beim nächsten a wieder auf die Geschwindigkeit 1 beschleunigt. Treffen beide Signale zur gleichen Zeit in der Mitte des Eingabewortes ein, dann ist das Wort in der Sprache L_{DD}.

Zum besseren Verständnis des Algorithmus wird der Lauf des Signals d_r durch das rechte Teilwort $D_\#(u_2)$ betrachtet. Ein solches Teilwort $D_\#(u_2)$ mit $|u_2| = k$ läßt sich in k Teilwörter $w_\nu, 1 \le \nu \le k$ der Länge $|w_\nu| = 2^{\nu-1}$ und ein #-Zeichen am rechten Rand aufteilen. Jedes w_ν besteht dabei aus $2^{\nu-1}-1$ #-Zeichen und einem Zeichen aus {a,b}. Im Beispiel u_2= abab ergeben sich die w_ν wie folgt:

$$D_\#(u_2) = a\#\#\#\#\#\#\#b\#\#\#a\#b\# = w_1 w_4 w_3 w_2 \# \text{ mit } w_1 = a,\ w_2 = \#b,$$
$$w_3 = \#\#\#a \text{ und } w_4 = \#\#\#\#\#\#\#b.$$

Die Geschwindigkeit des Signals d_r wird in jedem Teilwort w_ν allein durch das am rechten Rand von w_ν stehende Zeichen aus {a,b} bestimmt. Ist dieses ein a, so durchquert d_r das Wort w_ν mit der Geschwindigkeit 1. Handelt es sich um ein b, dann hat d_r in w_ν die Geschwindigkeit $\frac{1}{2}$, das Signal wird also in w_ν gegenüber dem ersten Fall um $|w_\nu| = 2^{\nu-1}$ Zeiteinheiten verzögert. Die Gesamtverzögerung von d_r im Teilwort $D_\#(u_2)$ gegenüber einem mit Einheitsgeschwindigkeit laufenden Signal ist die Binärzahl β, die sich ergibt, wenn man in u_2 jedes a durch eine 0 und jedes b durch eine 1 ersetzt. Damit ist jedes Teilwort $D_\#(u_2)$ eindeutig durch die Verzögerung β bestimmt.

Die Verzögerung des Signals d_l im vorderen Teilwort $(D_\#(u_1))^R$ ergibt sich entsprechend als die Binärzahl, die man durch Ersetzen der a's und b's in u_1 durch Nullen und Einsen erhält.

Beim Eingabewort $(D_\#(abab))^R XX D_\#(abab)$ ergibt sich beispielsweise eine Verzögerung der Signale d_r und d_l um jeweils 5 Takte:

abab $\longrightarrow$ 0101 $\simeq$ 5.

Insgesamt wird mit dem vorgestellten Verfahren nur eine konstante Zahl von Zustandsänderungen pro Zelle benötigt.

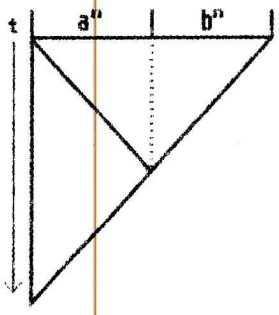
6.2 Zeitabhängige und zustandsabhängige Kommunikation

Der Unterschied zwischen den beiden im letzten Beispiel vorgestellten Verfahren zur Erkennung von L_{DD} besteht vor allem in der Funktion der durch den Zellularautomaten laufenden Signale:

Im ersten Verfahren wird für jedes a und jedes b des hinteren Teilwortes ein entsprechendes Signal nach links geschickt, wo der Zustand der Signale mit den im linken Teilwort stehenden Zeichen aus {a,b} verglichen wird. Beim zweiten Verfahren ist nicht der Zustand, sondern der Zeitpunkt des Eintreffens der Signale entscheidend für das Ergebnis. In dem einen Fall wird also Information zwischen verschiedenen Blöcken des Zellularautomaten durch den "Inhalt" eines Signals übertragen, im anderen Fall durch das Zeitverhalten.

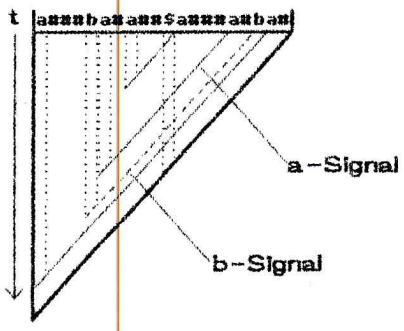
Diese beiden verschiedenen Möglichkeiten, Information durch einem Zellularautomaten zu schicken, werden im folgenden als zustandsabhängige bzw. zeitabhängige Kommunikation bezeichnet. Die Begriffe werden nochmal in den Abbildungen 6.3 a) und b) an einfachen Beispielen veranschaulicht. In komplizierteren Beispielen werden aber in der Regel beide Arten der Kommunikation kombiniert, um einen möglichst einfaches Erkennungsverfahren zu erhalten. Eine exakte Definition dieser Begriffe wird nicht gegeben; sie dürfte auch schwierig sein, da eine eindeutige Zuordnung der Signale eines gegebenen Zellularautomaten zu den verschiedenenen Kommunikationsarten nicht immer möglich scheint.

Abb. 6.3 *a) Zeitabhängige Kommunikation*



Bei der Erkennung von $\{a^n b^n \mid n \in \mathbb{N}\}$ mit dem abgebildeten Verfahren hängt daa Ergebnis nur vom Zeitpunkt des Eintreffens der vom linken und rechten Rand kommenden Signale ab, nicht jedoch vom Zustand der Signale, da aus jeder Richtung nur ein Signal in die Mitte läuft.

b) Zustandsabhängige Kommunikation



Im skizzierten Erkennungsverfahren für $\{w_1\$w_2 \mid (h_\#(w_2))^R = h_\#(w_1) \in \{a,b\}^+\}$ wird für jedes a und jedes b ein a- bzw. b-Signal nach links geschickt. Passiert ein solches Signal das $-Zeichen in der mutmaßlichen Mitte der Eingabe, so wird sein Zustand mit dem nächsten noch unmarklerten Zeichen aus {a,b} verglichen und das Zeichen anschließend markiert. Das Ergebnis der Erkennung hängt hier nur von den Zuständen der im linken Teilwort ankommenden Signale ab, nicht aber vom Zeitpunkt ihres Eintreffens - es wird lediglich vorausgesetzt, daß sie in der richtigen Reihenfolge eintreffen.

Das Beispiel der Sprache L_{DD} läßt erkennen, daß sich unter Ausnutzung von zeitabhängiger Kommunikation unter Umständen geringere Zustandsänderungszahlen erreichen lassen als nur mit zustandsabhängiger Kommunikation. Dies läßt sich auch mit Hilfe der Abschätzungen aus Lemma 3.4 für die maximale Anzahl der Verhaltensfolgeausschnitte in einem ρ-zustandsänderungsbeschränkten ZA verdeutlichen:

In einem Zellularautomaten kann an jede Zelle Information nur über die Zustände ihrer Nachbarzellen übertragen werden. Daher ist die Information, die die ν-te Zelle eines ZA von rechts bzw. links empfangen kann, abhängig von der Anzahl der möglichen Verhaltensfolgeausschnitte $\langle c_o(\nu+1)\rangle$ bzw. $\langle c_o(\nu-1)\rangle$. Die Anzahl dieser Verhaltensfolgeausschnitte wird daher im folgenden als Maß für die in einem ρ-änderungsbeschränkten ZA mögliche Kommunikation betrachtet, sie ist nach Lemma 3.1 und Lemma 3.4 a) beschränkt durch

$$\binom{\tau(n)}{\rho(n)} |A|^{\rho(n)+1}.$$

Dabei steht der erste Faktor für die Anzahl der Möglichkeiten, die $\rho(n)$ Zeitpunkte auszuwählen, an denen Zustandswechsel stattfinden. Der zweite Faktor entspricht den Auswahlmöglichkeiten für den Anfangszustand und die $\rho(n)$ neuen Zustände nach den Zustandswechseln. Damit liegt es nahe, den ersten Faktor als den von zeitabhängiger Kommunikation erzeugten Anteil anzusehen und den zweiten Faktor der zustandsabhängigen Kommunikation zuzuordnen.

Während der zustandsabhängige Anteil nur durch die Konstante $|A|$ und die Zustandsänderungsbeschränkung $\rho(n)$ bestimmt wird, ist der zeitabhängige Anteil auch noch von der Zeitbeschränkung $\tau(n)$ abhängig; für den Fall der Beschränkung auf Realzeit ergibt er sich als

$$\binom{\tau(n)}{\rho(n)} = \binom{n}{\rho(n)} = O(n^{\rho(n)}) = O(|A|^{c \cdot \mathrm{ld}\, n \cdot \rho(n)}) \quad \text{mit } c = \frac{1}{\mathrm{ld}\,|A|},$$

gegenüber dem zustandsabhängigen Anteil

$$|A|^{\rho(n)+1} = O(|A|^{\rho(n)}).$$

In einem ρ-zustandsänderungsbeschränkten Zellularautomaten ließe sich demnach mit zeitabhängiger Kommunikation mehr Information übertragen als mit zustandsabhängiger. So ist zum Beispiel der zeitabhängige Anteil in einem $O(1)$-änderungsbeschränkten ZA von derselben Größenordnung wie der zustandsabhängige Anteil in einem $O(\ln n)$-änderungsbeschränkten ZA, diese Anschauung wird auch durch die beiden am Anfang des Kapitels vorgestellten Algorithmen für L_{DD} bestätigt.

Beschränkt man sich in einem $O(1)$-änderungsbeschränkten ZA auf zustandsabhängige Kommunikation, so werden offenbar nur reguläre Sprachen erkannt. Die gesamte Konstruktion der Sprachklassen mit Zustandsänderungskomplexitäten $\ln_{(k)} n$, $k>1$ in Abschnitt 3.3 beruht in entscheidenden Bereichen auf der Verwendung zeitabhängiger Kommunikation, wie zum Beispiel bei der Erkennung von Wörtern der Struktur $D_{\#}(w)$. Vermutlich ist in Zellularautomaten mit Zustandsänderungszahlen kleiner $O(\ln n)$ die zeitabhängige Kommunikation notwendig, um irgendwelche nichtregulären Sprachen zu erkennen.

Um für eine gegebene Sprache ein Verfahren mit möglichst wenigen Zustandsänderungen zu bekommen, wird man daher beim Entwurf dieses Verfahrens häufig Gebrauch von den Möglichkeiten der zeitabhängigen Kommunikation machen. Überträgt man aber dann den so entworfenen Algorithmus auf die in Kapitel 5 beschriebenen SZA in der Hoffnung, dadurch auch eine minimale oder wenigstens eine günstige Pipeline-Periode zu erhalten, so wird man in der Regel enttäuscht. Ein gutes Beispiel dafür ist die Erkennung der Sprache $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ in einem SUZA aus Beispiel 5.1.

Dort wird ein sehr einfaches, $O(1)$-änderungsbeschränktes Verfahren zur Erkennung von $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ auf einen SUZA übertragen, wobei man eine Pipeline-Periode von $O(n)$ erhält. Mit einem anderen Algorithmus unter Verwendung von Zählern erhält man mit $O(\ln n)$ Zustandsänderungen pro Zelle auch eine Pipeline-Periode von $O(\ln n)$. Der Grund dafür ist offensichtlich, daß das erste Verfahren auf zeitabhängiger, das zweite dagegen auf zustandsabhängiger Kommunikation beruht. Ein SZA kann aber zeitabhängige Kommunikation in der Regel nur mit einer Pipeline-Periode von $O(n)$ durchführen.

Dies läßt sich auch mit der Anzahl der Verhaltensfolgeausschnitte $\langle c_o(v) \rangle$ in einem SZA begründen. In einem ρ-SZA gibt es nicht mehr als

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

solche Verhaltensausschnitte. Vergleicht man diesen Term mit dem entsprechenden Term eines ρ-änderungsbeschränkten Zellularautomaten, so stellt man fest, daß hier genau der zeitabhängige Anteil fehlt.

Die verschiedenen Erkennungsmächtigkeiten von änderungsbeschränkten UZA und SUZA sind anscheinend in erster Linie durch das Fehlen der zeitabhängigen Kommunikation in den SUZA bedingt; beim Vergleich der änderungsbeschränkten bidirektionalen ZA mit den SZA kommt noch hinzu, daß in SZA Informationen nur über geringe Entfernungen von links nach rechts übertragen werden können. In diesem Zusammenhang ist es nicht überraschend, daß zur Erkennung nichtregulärer Sprachen in SUZA und SZA eine Pipeline-Periode von $O(\ln n)$ benötigt wird.

Ein Algorithmus mit einer kleinen Anzahl von Zustandsänderungen ist also nicht unbedingt günstig, um bei Übertragung auf SZA auch eine kleine Pipeline-Periode zu erreichen. Es kommt vielmehr auf die Art der im Algorithmus stattfindenden Kommunikation an, zeitabhängige Signale machen die Berechnung ziemlich "sperrig"; es werden dann mit dem Verfahren aus Abschnitt 5.4 mehr Zellen benötigt, um eine konstante Pipeline-Periode zu erreichen. Algorithmen, die auf der Verwendung zustandsabhängiger Kommunikation beruhen, sind dagegen "flexibler" auch insofern, als sich das Raum-Zeit-Diagramm einer solchen Berechnung in einem gewissen Rahmen verformen und verzerren läßt, ohne daß dadurch das "Timing" des Verfahrens durcheinandergerät. Ein gutes Beispiel für diese Eigenschaft ist die im nächsten Kapitel noch kurz angesprochene Superstabilität.

7. Weitere Ergebnisse und Möglichkeiten

Zum Schluß der Arbeit werden nun noch einige Bereiche behandelt, die in den vorangegangenen Kapiteln nur am Rande betrachtet wurden. Es werden Teilergebnisse angegeben und noch offene Fragen vorgestellt.

7.1 Sum-zustandsänderungsbeschränkte Zellularautomaten

Einige der in dieser Arbeit bewiesenen Ergebnisse über max-änderungsbeschränkte ZA können recht einfach auf die hier bisher kaum betrachteten sum-änderungsbeschränkten Zellularautomaten übertragen werden. Dazu zählen insbesondere die in den Abschnitten 3.2 und 3.3 nachgewiesenen Hierarchien der Sprachklassen und die Abschlußeigenschaften bezüglich Spiegelbild aus Abschnitt 3.5.

Aus den Definitionen der max- und sum-Zustandsänderungsbeschränkung folgt unmittelbar, daß jeder ρ-max-änderungsbeschränkte ZA $\rho \cdot n$-sum-änderungsbeschränkt ist. In [Vo82b] wurde bewiesen, daß die Umkehrung nicht gilt. Trotzdem kann man mit den gleichen Sprachen, mit denen in Kapitel 3 die Hierarchie der max-änderungsbeschränkten ZA nachgewiesen wurde, eine ähnliche Hierarchie für die sum-änderungsbeschränkten ZA beweisen.

Die in Abschnitt 3.3 vorgestellten Algorithmen für $L_{RQ_{(k)}}$ und $L_{RD_{(k)}}$ benötigen $O(n \cdot n^{\frac{1}{2^k}})$- bzw. $O(n \cdot \ln_{(k)} n)$ Zustandsänderungen im Gesamtautomaten. Um zu zeigen, daß diese Sprachen jeweils nicht mit einer wesentlich kleineren sum-Änderungszahl erkannt werden, bedient man sich eines Beweisschemas aus [Vo82a], mit dem dort gezeigt wird, daß die Sprache $L_R = \{uYYu^R \mid u \in \{a,b\}^+\}$ nicht mit einer sum-Änderungsbeschränkung von $O(n^{2-\varepsilon})$ erkannt werden kann.

Auch die Abschlußeigenschaften bezüglich Spiegelbild können mit denselben Sprachen wie in Abschnitt 3.5 nachgewiesen werden. Dazu muß Satz 3.3 entsprechend an die sum-änderungsbeschränkten ZA angepaßt werden.

7.2 Unäre Sprachen

Unidirektionale SZA scheinen eine einfache Möglichkeit zu bieten, die von Zellularautomaten erkannten unären Sprachen, also Sprachen über einem einelementigen Alphabet, besser zu strukturieren und dadurch vielleicht einige offene Probleme in der Zeithierarchie der Zellularautomaten zu lösen oder wenigstens besser zu verstehen.

In [Se79] wurde bewiesen, daß unidirektionale Zellularautomaten nur die regulären unären Sprachen in Realzeit erkennen. In Linearzeit lassen sich aber auch einige nicht reguläre unäre Sprachen von UZA erkennen. So wurde in Beispiel 5.1 gezeigt, wie die Sprache $\{a^{2^n} \mid n \in \mathbb{N}\}$ von einem $O(\ln n)$-SUZA erkannt werden kann. Es ist jedoch noch nicht bekannt, welche Zeitkomplexitäten zwischen den Mengen der in Realzeit und der in Linearzeit erkannten Sprachen es noch gibt.

Eine Sprache, die von einem ρ-SUZA erkannt werden kann, kann natürlich auch von einem UZA in der Zeit $n+\rho(n)$ erkannt werden. Im Fall der unären Sprachen läßt sich aber auch die Umkehrung zeigen. Es gilt für unäre Sprachen

$$L \in SUZA(\rho) \iff L \in UZA_{n+\rho}.$$

Dieses Ergebnis ergibt sich aus der Tatsache, daß in einem Realzeit-UZA bei der Erkennung einer unären Sprache die eigentliche Berechnung von einem einzigen, vom rechten Rand kommenden Signal geleistet wird. Im Innern des oberen, linken Dreiecks des Raum-Zeit-Diagramms passiert also "nichts wesentliches", darauf beruht ja auch der Beweis dafür, daß in Realzeit-UZA nur reguläre Sprachen erkannt werden können. Ein SUZA ist aber genau ein UZA, in dem dieser Teil der Berechnung weggelassen wird.

Um zu zeigen, daß es Sprachen gibt, die von UZA in Linearzeit, aber nicht in der Zeit $n+\ln n$ erkannt werden, reicht es daher aus, die Existenz einer unären Sprache aus UZA_{2n} zu zeigen, die von keinem $O(\ln n)$-SUZA erkannt werden kann. Also ist mindestens eine der folgenden Aussagen richtig:

Alle unären Sprachen aus $SUZA(O(n))$ sind bereits in $SUZA(O(\ln n))$

oder

es gilt $UZA_R \subset UZA_{n+\ln n} \subset UZA_{2n}$.

Aufgrund der Äquivalenz von SUZA und Pseudo-Realzeit-UIA können diese Fragen auch mit Hilfe der UIA und der die UIA charakterisierenden sequentiellen Maschinen untersucht werden.

Weitere offene Fragen zu den unären Sprachen, die sich im Zusammenhang mit dieser Arbeit stellen, sind unter anderem :

- Sind alle unären Sprachen aus ZA_R oder ZA_{2n} mit einer konstanten Anzahl von Zustandsänderungen erkennbar ?
- Kann man insbesondere die Sprache $\{a^p \mid p \text{ prim}\}$ mit einer konstanten Anzahl von Zustandsänderungen pro Zelle oder mit Pipeline-Periode $O(\ln n)$ erkennen ?
- Die Sprachen $\{a^{2^{(k)n}} \mid n \in \mathbb{N}\}$ lassen sich in einem $O(\ln n)$-SZA durch ein mit Binärzählern arbeitendes Verfahren erkennen, damit sind diese Sprachen in der Klasse $ZA_{n+\ln n}$. Sind diese Sprachen in Realzeit erkennbar ?

7.3 Superstabilität

Die im letzten Kapitel eingeführte Einteilung in zeit- und zustandsabhängige Kommunikation kann auch mit einem anderen, bekannten Problem in der Theorie der Polyautomaten veranschaulicht werden, nämlich der Frage nach der in [CGS86] untersuchten Superstabilität von Sprachen.

Definition 7.1 a) **Eine Sprache** $L \subseteq (V \cup \{\#\})^+$ **heißt superstabil, wenn für alle w aus** $(V \cup \{\#\})^+$ **gilt:** $w \in L \iff h_{\#}(w) \in L$.

b) **Es sei** $L \subseteq V^+$ **und** $\# \notin V$. **Dann heißt**

$$S_{\#}(L) := \{ w \in (V \cup \{\#\})^+ \mid h_{\#}(w) \in L \}$$

die zu L gehörende superstabile Sprache.

Eine superstabile Sprache ist also invariant gegenüber dem Einfügen von #-Zeichen. Ist w ein Wort aus einer superstabilen Sprache, so sind auch alle Wörter, die man durch Einfügen oder Löschen einer beliebigen Anzahl von # erhält, in der Sprache enthalten. In [CGS86] wurde gezeigt, daß die Menge UZA_R abgeschlossen bezüglich der Abbildung $S_{\#}$ ist, für bidirektionale ZA ist diese Frage noch offen.

Versucht man, aus einem vorhandenen Algorithmus für eine Sprache L einen "superstabilen" Algorithmus für $S_{\#}(L)$ zu machen, so stellt man fest, das dies für Algorithmen, die nur zustandsabhängige Kommunikation verwenden, wesentlich einfacher ist als für solche, die auf zeitabhängiger Kommunikation beruhen. Dies erklärt sich daraus, daß die durch eingefügte #-Zeichen hervorgerufenen längeren Laufzeiten der Signale das "Timing" der zeitabhängigen Kommunikation völlig durcheinanderbringen. Bei Algorithmen mit zustandsabhängiger Kommunikation macht das Einfügen von zusätzlichen #-Zeichen gegenüber der ursprünglichen Sprache L jedoch kaum einen Unterschied im Ablauf der Erkennung, da die Reihenfolge des Eintreffens der Signale in den Zellen ja gleich bleibt.

Die Frage nach der Abgeschlossenheit der Klasse ZA_R bezüglich $S_{\#}$ ist also auch eine Frage nach den "Kosten" für einen Verzicht auf zeitabhängige Kommunikation.

Es ist auch noch nicht bekannt, ob die Sprachklassen der SZA und der änderungsbeschränkten ZA abgeschlossen bezüglich $S_{\#}$ sind, dem betrachteten Modell zufolge sollte dies für $\rho < O(n)$ jedoch höchstens für die Klassen SZA(ρ) und SUZA(ρ) möglich sein, nicht jedoch für die Sprachklassen der änderungsbeschränkten ZA.

8. Literaturverzeichnis

[BuC84] W. Bucher, K. Culik II
On real-time and linear-time cellular automata
RAIRO informatique theorique, Vol. 18, No. 4 (1984), 307-325

[Co69] S. N. Cole
Real-time computation by n-dimensional iterative arrays of finite state machines
IEEE Transactions on Computers, Vol. C-18, No. 4 (1969), 349-365

[CGS84] K. Culik II, J. Gruska, A. Salomaa
systolic trellis automata, part I
Int. Journal of Computer Mathematics, Vol. 15, (1984), 195-212

[CGS86] K. Culik II, J. Gruska, A. Salomaa
Systolic trellis automata : Stability, decidability and complexity
Information and Control, Vol. 71 (1986), 218-230

[ChC85] C. Choffrut, K. Culik II
On real-time cellular automata and trellis automata
Acta Informatica, No. 21 (1984), 393-407

[CIV88] J. H. Chang, O. H. Ibarra, A. Vergis
On the power of one-way communication
Journal of the ACM, Vol. 35, No. 3 (1988), 697-726

[CSW84] K. Culik II, A. Salomaa, D. Wood
Systolic tree acceptors
RAIRO informatique theorique, Vol. 18, No. 1 (1984), 53-69

[CuY84] K. Culik II, S. Yu
Iterative tree automata
Theoretical Computer Science, Vol. 32 (1984), 227-247

[Gru84] J. Gruska
Systolic automata - power, characterizations, nonhomogeneity
Proc. of MFCS'84, Lecture Notes in Computer Science 233, 140-153

[IbJ87] O. H. Ibarra, T. Jiang
On one-way cellular arrays
SIAM Journal of Computing, Vol. 16, No. 4 (1987), 1135-1154

[IPK85] O. H. Ibarra, M. A. Palis, S. M. Kim
Fast parallel language recognition by cellular automata
Theoretical Computer Science, Vol. 41 (1985), 231-246

[Se79] S. R. Seidel
Language recognition and the synchronization of cellular automata
Dissertation, Univ. of Iowa (1979)

[SoW83] R. Sommerhalder, S. van Westrhenen
Parallel language recognition in constant time by cellular automata
Acta Informatica, Vol. 19 (1983), 397-407

[Vo79] R. Vollmar
Algorithmen in Zellularautomaten
Teubner, Stuttgart (1979)

[Vo81] R. Vollmar
On cellular automata with a finite number of state changes
Computing, Suppl. 3 (1981), 181-191

[Vo82a] R. Vollmar
Some remarks about the "efficiency" of polyautomata
Int. Journal of Theoretical Physics, Vol. 21, No. 12 (1982), 1007-1015

[Vo82b] R. Vollmar
Zur Zustandsänderungskomplexität von Zellularautomaten
Informatik-Skripten 2, TU Braunschweig (1982), 139-151

[Vo84] R. Vollmar
Einige Bemerkungen zur Zustandsänderungskomplexität von Polyautomaten
Informatik-Skripten 6, TU Braunschweig (1984), 146-151

[Vo87] R. Vollmar
Some remarks on pipeline processing by cellular automata
Computers and Artificial Intelligence, Vol. 6, No. 3 (1987), 263-278