Algorithmen in Zellularautomaten
Aufgabenblatt 6

Aufgabe 6.1
Zeigen Sie: Wenn man von einem Signal für Konstruierbarkeit nur fordert, dass es «nicht springt» und «leicht erkennbar» ist, dann ist *jedes* solche Signal von einem ZA konstruierbar.

Lösung 6.1
Man nutzt aus, dass beliebige Anfangskonfigurationen erlaubt sind.

Wir benutzen 8 Zustände: $Q = \{$ `*`, `<*`, `*>`, `L`, `LR`, `R`, `RL`, `__` $\}$.

- `__` ist Ruhezustand.
- Die Zustände `*`, `<*` und `*>` markieren die Position des Signals.
- Die Zustände `R` und `RL` wanderen nach links und schieben die Signalmarkierung nach rechts bzw. nach rechts und anschließend sofort wieder nach links.
- Die Zustände `L` und `LR` wanderen nach rechts und schieben die Signalmarkierung nach links bzw. nach links und anschließend sofort wieder nach rechts.

Man mache sich klar, dass sich bei geeigneter Wahl der Überführungsfunktion z. B. folgendes Raum-Zeit-Diagramm ergibt:

```
LR  __  LR  __  __  __  *   __  R   __  R   __  RL  __
__  LR  __  LR  __  __  *   R   __  R   __  RL  __  __
__  __  LR  __  LR  __  __  *   R   __  RL  __  __  __
__  __  __  LR  __  LR  __  __  *   RL  __  __  __  __
__  __  __  __  LR  __  LR  __  __  <*  __  __  __  __
__  __  __  __  __  LR  __  LR  *   __  __  __  __  __
__  __  __  __  __  __  LR  *>  __  __  __  __  __  __
__  __  __  __  __  __  __  LR  *   __  __  __  __  __
__  __  __  __  __  __  __  *>  __  __  __  __  __  __
__  __  __  __  __  __  __  __  *   __  __  __  __  __
__  __  __  __  __  __  __  __  *   __  __  __  __  __
__  __  __  __  __  __  __  __  *   __  __  __  __  __
```

Aufgabe 6.2
Geben Sie möglichst präzise einen Zellularautomaten an, der die Markierungen $\{(1, i^3) \mid \forall i \in \mathbb{N}_+\}$ realisiert.

Lösung 6.2
Grundsätzliche Idee: benutze die Rekursion $(i+1)^3 = i^3 + 3i^2 + 3i + 1$.

Analog zum Parabelsignal in der Vorlesung verbringe man Zeit $3i^2 + 3i + 1$ bevor man das Signal von Position i zu Position $i + 1$ verschiebt.

Aufgabe 6.3
Geben Sie möglichst präzise einen Zellularautomaten an, der jede Konfiguration der Form

	(	.	.	.	.	.	.	.	)	

überführt in die korrespondierende Konfiguration

	(	.	x	.	.	.	.	.	)	

wobei die Länge des Abschnittes $(\cdots$x gerade die Wurzel der Länge des Abschnittes $(\cdots)$ sein soll.

Lösung 6.3
Man sende vom linken Ende ein Parabelsignal und vom rechten Ende ein Signal mit Geschwindigkeit 1. Zwei Beispiele machen hoffentlich alles klar:

	1	2	3	4	5	6	7	8	9
$t = 1$:	(	.	.	.	.	.	.	.	)
$t = 2$:	(|	.	.	.	.	.	.	<	)
$t = 3$:	(	<|	.	.	.	.	<	.	)
$t = 4$:	(<	|	.	.	.	<	.	.	)
$t = 5$:	(>	|	.	.	<	.	.	.	)
$t = 6$:	(	>|	.	<	.	.	.	.	)
$t = 7$:	(	.	**	.	.	.	.	.	)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$t = 1$:	(	.	.	.	.	.	.	.	.	.	.	.	.	.	.	)
$t = 2$:	(|	.	.	.	.	.	.	.	.	.	.	.	.	.	<	)
$t = 3$:	(	<|	.	.	.	.	.	.	.	.	.	.	.	<	.	)
$t = 4$:	(<	|	.	.	.	.	.	.	.	.	.	.	<	.	.	)
$t = 5$:	(>	|	.	.	.	.	.	.	.	.	.	<	.	.	.	)
$t = 6$:	(	>|	.	.	.	.	.	.	.	.	<	.	.	.	.	)
$t = 7$:	(	.	<|	.	.	.	.	.	.	<	.	.	.	.	.	)
$t = 8$:	(	<	|	.	.	.	.	.	<	.	.	.	.	.	.	)
$t = 9$:	(<	.	|	.	.	.	.	<	.	.	.	.	.	.	.	)
$t = 10$:	(>	.	|	.	.	.	<	.	.	.	.	.	.	.	.	)
$t = 11$:	(	>	|	.	.	<	.	.	.	.	.	.	.	.	.	)
$t = 12$:	(	.	>|	.	<	.	.	.	.	.	.	.	.	.	.	)
$t = 13$:	(	.	.	**	.	.	.	.	.	.	.	.	.	.	.	)

Aufgabe 6.4
Überlegen Sie sich eine Konstruktion möglichst ähnlich zu den wandernden Zählern aus der Vorlesung, bei der aber immer das *höchstwertige* Bit vorausläuft.

Lösung 6.4
Eine mögliche Idee:

Das niedrigstwertige Bit «schiebt anderen Bits vor sich her», bewegt sich aber nur noch mit Geschwindigkeit 1/2. Das erlaubt den Überträgen, mit Geschwindigkeit 1 zu den höherwertigen Bits «nach vorne» zu laufen. In Abbildung 6.1 ist ein Beispiel dargestellt.

Aufgabe 6.5
Ein wandernder Zähler kann auch als Algorithmus für die Umwandlung von Unär- in Binärdarstellung angesehen werden. Überlegen Sie sich einen Zellularautomaten, der die umgekehrte Umwandlung vornimmt.

Lösung 6.5
Man benutzt einen Wanderzähler, der herunter zählt.

Das geht nicht ganz so «schön»/schnell wie heraufzählen, denn (wenn das niedrigstwertige Bit vorausläuft) ist das Identifizieren der Situation, wenn der

Zähler auf 0 ist, nicht ganz so einfach. Ein möglicher Ausweg: Das niedrigstwertige Bit bewegt sich nur mit Geschwindigkeit 1/2, «zieht den Rest hinter sich her» oder «schiebt den Rest vor sich her» und der Rest kann schrumpfen, wenn das höchstwertige Bit von 1 auf 0 wechselt, also der Zähler um eine Stelle kürzer wird.

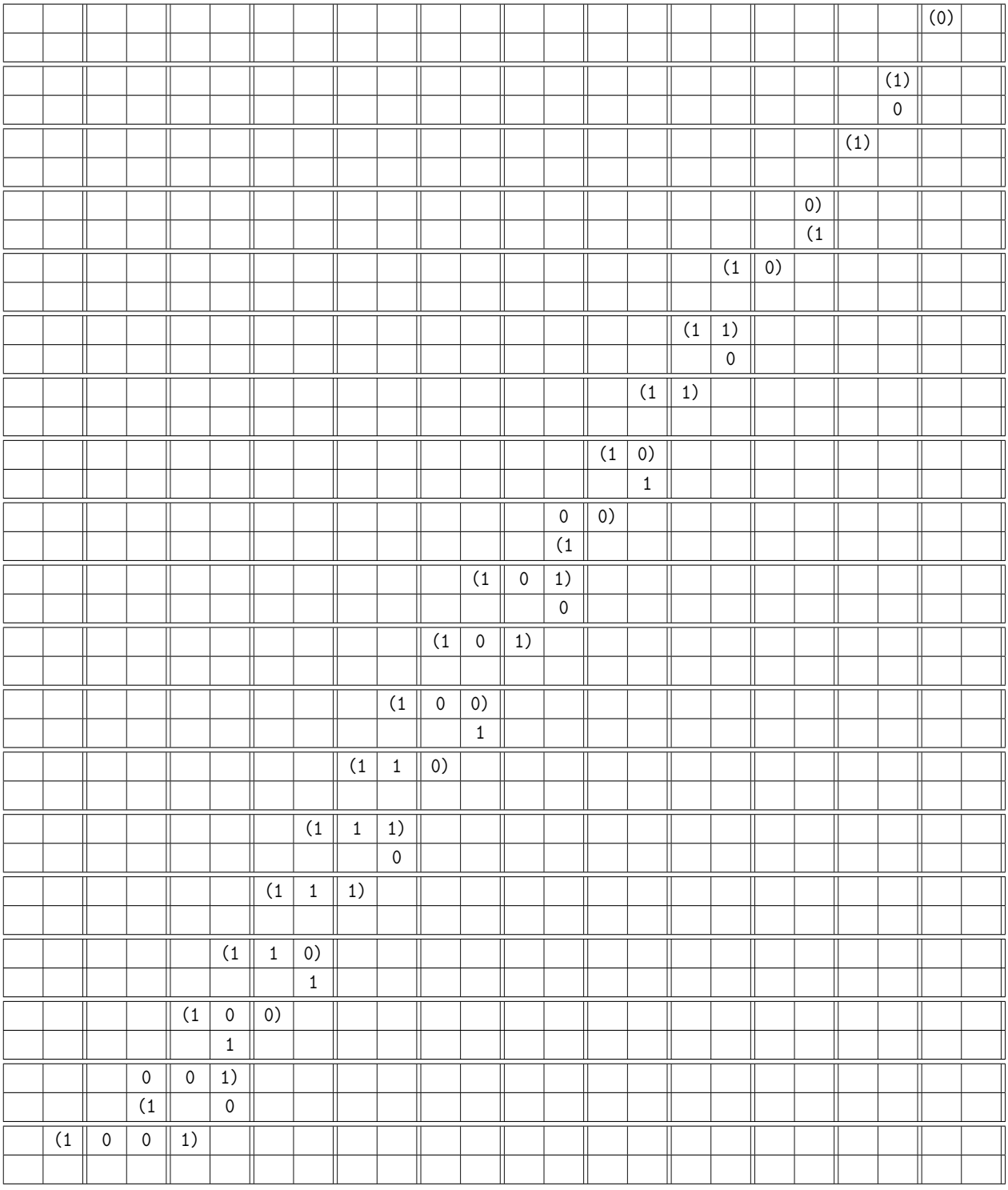

Abbildung 6.1: Wanderzähler nach links mit *höchstwertigem* Bit voraus. Das niedrigstwertige Bit «schiebt» mit Geschwindigkeit 1/2 den Rest vor sich her.