

# Algorithmen in Zellularautomaten

## Aufgabenblatt 7

### Aufgabe 7.1

Machen Sie sich Gedanken zum eindimensionalen FSSP mit zwei Generälen:

1. Präzisieren Sie die Problemstellung.
2. Was ist eine gute untere Schranke für den Zeitbedarf?
3. Überlegen Sie sich einen möglichst schnellen Algorithmus.

### Lösung 7.1

1. Anfangskonfigurationen enthalten *genau zwei* Zellen im Zustand  $g$
2. Es sei  $k_l$  die Anzahl der Zellen im Zustand  $s$  links vom linken General und analog  $k_r$  die Anzahl der Zellen im Zustand  $s$  rechts vom rechten General. Zwischen den beiden Generälen seien  $k_m$  Zellen.

Insgesamt sind also  $n = k_l + 1 + k_m + 1 + k_r$  Zellen zu synchronisieren.

Es sei  $m = \max(k_l, k_r)$ . Dann dauert es  $m + n - 1$  Schritte bis zum ersten Mal die Randzelle, die näher an einem General liegt, ein Signal vom anderen Rand erhalten haben kann.

Das ist jedenfalls eine untere Schranke ... und wie man in Teil 3 sieht, gibt es tatsächlich Algorithmen, die so schnell sind.

3. Idee: Es sei  $M$  die Zelle in der Mitte zwischen den beiden Generälen. Man sorgt dafür, dass sich der ZA im wesentlichen so verhält wie der für nur einen General an der Stelle  $M$  ab dem Zeitpunkt, da jener  $(k_r - M)/2$  Schritte gemacht hat; siehe Abbildung 7.1.

### Aufgabe 7.2

Machen Sie sich Gedanken zum eindimensionalen FSSP mit beliebig vielen Generälen:

1. Präzisieren Sie die Problemstellung.
2. Was ist eine gute untere Schranke für den Zeitbedarf?
3. Überlegen Sie sich einen möglichst schnellen Algorithmus.

### Lösung 7.2

1. In der Anfangskonfiguration gibt es eine beliebige positive Anzahl von Zellen im Zustand  $g$ ; links und rechts von jedem eine beliebige Zahl von Zellen im Zustand  $s$ .

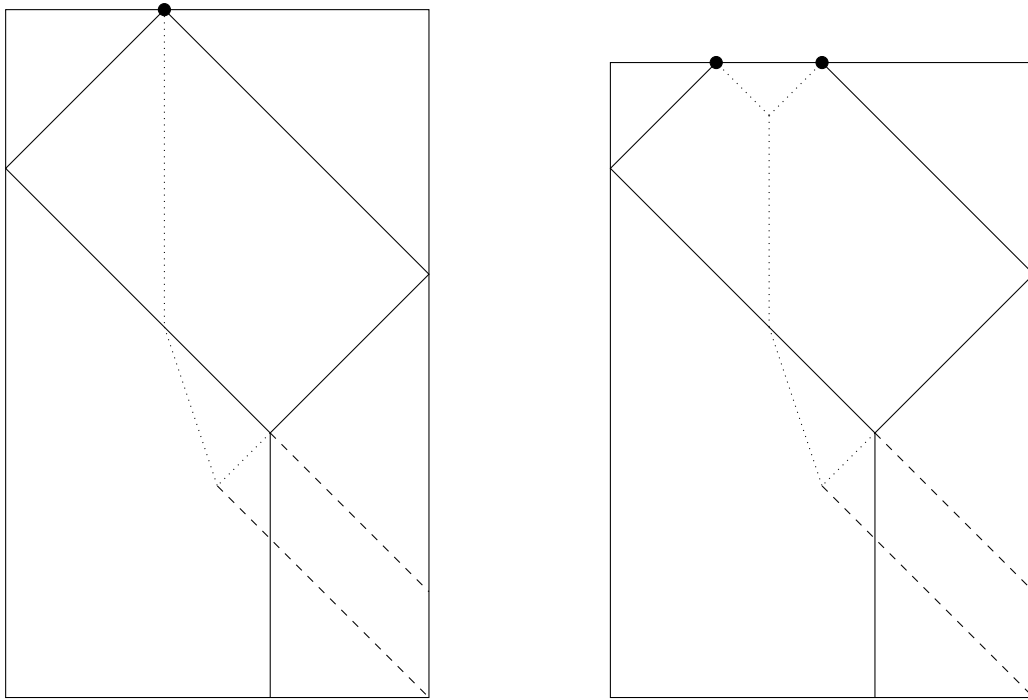


Abbildung 7.1: Synchronisation mit einem General (links) und mit zwei Generalen (rechts).

2. Mit dem „üblichen“ Argument, dass jeder Rand „etwas vom anderen Rand wissen muss“ ergibt sich, dass für eine untere Schranke nur die Position  $k_l$  des am weitesten links und die Position des am weitesten rechts positionierten Generals  $k_r$  Zellen vom rechten Rand relevant sind. Untere Schranke wie in Aufgabe 7.1:  $\max(k_l, k_r) + n - 1$  Schritte. Und das kann man erreichen.
3. Die Herleitung der unteren Schranke nutzt nur die beiden äußeren Generäle. Ein entsprechender Algorithmus letztendlich ebenfalls: Man kann rechtzeitig alle Signale «unschädlich machen», die von den Generalen dazwischen gestartet werden.

### Aufgabe 7.3

In dieser Aufgabe geht es um das FSSP in  $d$ -dimensionalen Quadern mit einem General in einer Ecke (Ihrer Wahl).

1. Was ist eine gute untere Schranke für den Zeitbedarf?
2. Überlegen Sie sich einen Algorithmus für den Fall  $d = 3$ .
3. Wie schnell ist Ihr Algorithmus?

### Lösung 7.3

Siehe Umeo/Nishide/Kubo (2012): A Simple Optimum-Time FSSP Algorithm for Multi-Dimensional Cellular Automata, JAC.

### Aufgabe 7.4

In dieser Aufgabe geht es um das FSSP in  $d$ -dimensionalen (Hyper-)Würfeln mit einem General in einer Ecke (Ihrer Wahl).

1. Was ist eine gute untere Schranke für den Zeitbedarf?
2. Überlegen Sie sich einen Algorithmus für den Fall  $d = 2$ .
3. Überlegen Sie sich einen Algorithmus für den Fall  $d = 3$ .
4. Wie schnell ist Ihr Algorithmus?

### Lösung 7.4

1. ...
2. Wie bei Quadraten. Aber nun ist sicher, dass der General «in der Mitte» des Winkels steht, also funktioniert Synchronisation in Zeit  $2n$ , wenn  $n$  die Seitenlänge des Quadrates ist. Die Synchronisation des nächsten Winkels muss 2 Schritte später gestartet werden; kein Problem.
3. knifflig; hat jemand eine schöne Idee?
4. nicht ohne Algorithmus

### \* Aufgabe 7.5

In dieser Aufgabe geht es um das FSSP in  $d$ -dimensionalen Quadern mit einem General an einer beliebigen Stelle.

1. Was ist eine gute untere Schranke für den Zeitbedarf?
2. Überlegen Sie sich einen Algorithmus für diese Problemvariante.
3. Wie schnell ist Ihr Algorithmus?

### Lösung 7.5

Siehe Umeo/Nishide/Kubo (2012): A Simple Optimum-Time FSSP Algorithm for Multi-Dimensional Cellular Automata, JAC.

### \* Aufgabe 7.6

In dieser Aufgabe geht es um das FSSP in  $d$ -dimensionalen (Hyper-)Würfeln mit einem General an einer beliebigen Stelle.

1. Was ist eine gute untere Schranke für den Zeitbedarf?
2. Überlegen Sie sich einen Algorithmus für diese Problemvariante.

3. Wie schnell ist Ihr Algorithmus?

### Lösung 7.6

1. ... [mühsam aufzuschreiben] ...
2. Einfache Algorithmen: Ein Signal in eine Ecke schicken und von dort Würfel synchronisieren oder einfach den Algorithmus für Quader nutzen.  
Das sind aber beides Algorithmen, die *nicht* die beste bekannte untere Zeitschranke erreicht.
3. Für die einfachen Algorithmen ist das banal.

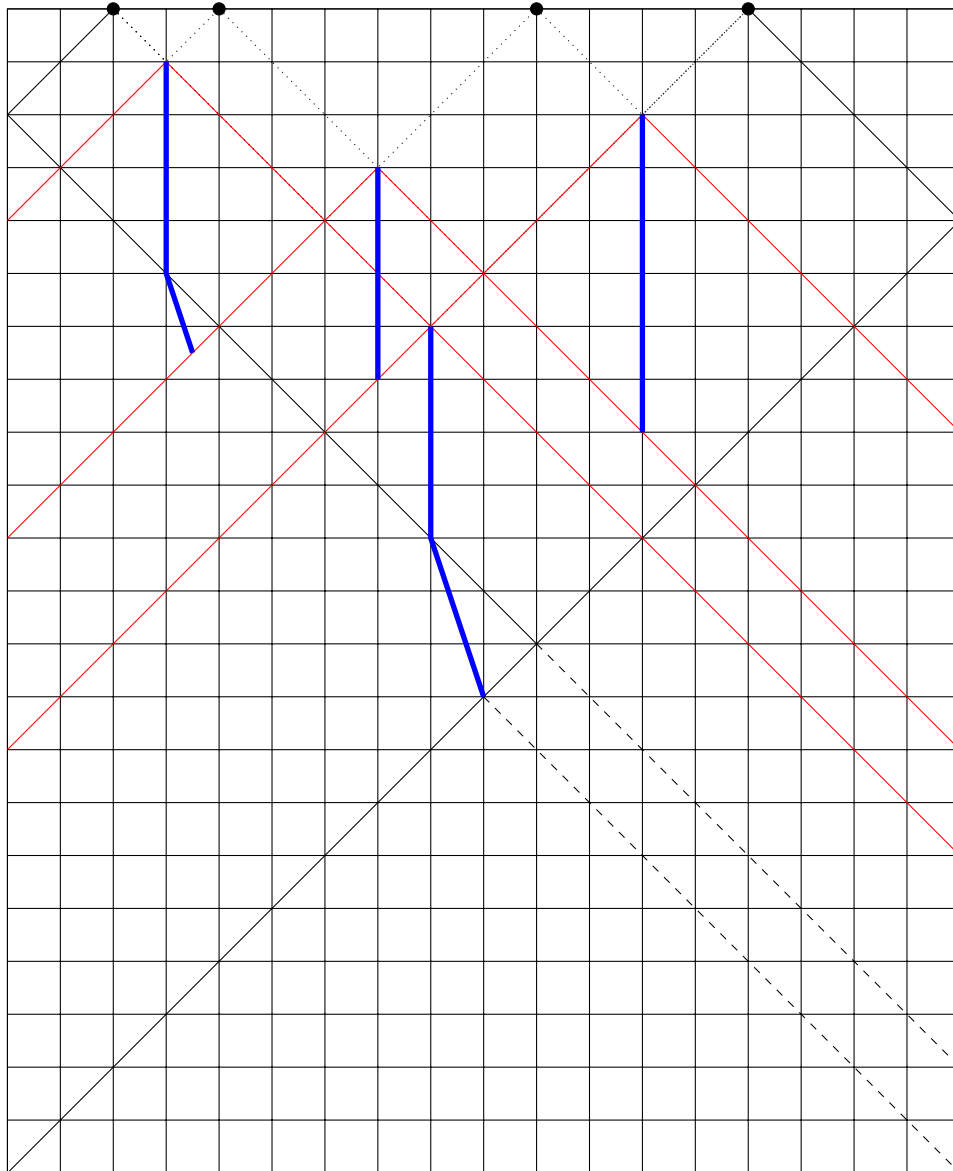


Abbildung 7.2: Skizze zur Synchronisation mit «vielen» Generälen.