



Numerische Simulationsverfahren im Ingenieurwesen

Hausaufgabenblatt 1

WS 17/18

Thema: Eindimensionale, zelluläre Automaten

Theorieaufgabe: Update-Regeln

Bearbeiten Sie nachfolgende Aufgabenstellungen zu den Update-Regeln zellulärer Automaten.

- a) Gegeben sei ein eindimensionaler Binärautomat mit dem Nachbarschaftsradius $r = 1$. Prüfen Sie, ob es sich bei der **Wolfram-Regel Nr. 102** um eine **additive Regel** handelt.
- b) Gegeben sei ein eindimensionaler Binärautomat mit dem Nachbarschaftsradius $r = 1$. Die Update-Regel ist durch

$$a_i(t+1) = g(s_i) = -\frac{1}{12} s_i (s_i - 1)(s_i + 1)(s_i + 3)(s_i + 4), \quad (1)$$

mit

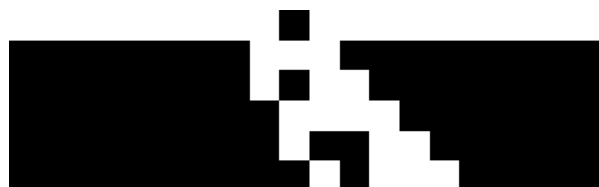
$$s_i(t) := -2a_{i-1}(t) + a_i(t) - 2a_{i+1}(t) \quad (2)$$

formuliert.

Welche Wolfram-Regel verbirgt sich hinter obiger Update-Regel? Begründen Sie!

- c) Die nachfolgende Raum-Zeit-Charakteristik gehört zu einem eindimensionalen Binärautomaten mit dem Nachbarschaftsradius $r = 1$. In der Anfangskonfiguration ist nur eine Zelle schwarz, alle anderen weiß, periodische Randbedingungen wurden genutzt. Welche Wolfram-Regel wurde als Update-Regel verwendet? Begründen Sie!

Raum-Zeit-Charakteristik eines einfachen Binärautomaten



- d) Gegeben ist ein eindimensionaler Automat mit dem Nachbarschaftsradius $r = 1$, dessen Zellen $k = 3$ verschiedene Zustände annehmen können, d.h. $a_i(t) \in \{0,1,2\}$. Die Update-Regel ist (für alle Zustandskombinationen der Nachbarschaft) nachfolgend aufgeführt:

$(0,0,0) \rightarrow 0$	$(1,0,0) \rightarrow 0$	$(2,0,0) \rightarrow 2$
$(0,0,1) \rightarrow 0$	$(1,0,1) \rightarrow 0$	$(2,0,1) \rightarrow 2$
$(0,0,2) \rightarrow 2$	$(1,0,2) \rightarrow 2$	$(2,0,2) \rightarrow 2$
$(0,1,0) \rightarrow 0$	$(1,1,0) \rightarrow 0$	$(2,1,0) \rightarrow 0$
$(0,1,1) \rightarrow 0$	$(1,1,1) \rightarrow 0$	$(2,1,1) \rightarrow 0$
$(0,1,2) \rightarrow 0$	$(1,1,2) \rightarrow 0$	$(2,1,2) \rightarrow 0$
$(0,2,0) \rightarrow 1$	$(1,2,0) \rightarrow 1$	$(2,2,0) \rightarrow 1$
$(0,2,1) \rightarrow 1$	$(1,2,1) \rightarrow 1$	$(2,2,1) \rightarrow 1$
$(0,2,2) \rightarrow 1$	$(1,2,2) \rightarrow 1$	$(2,2,2) \rightarrow 1$

Vereinfachen Sie obige Tabelle, indem Sie als Platzhalter für einen beliebigen, möglichen Zustandswert einen Punkt „•“ nutzen, also beispielsweise, $(\bullet, 1, 1) \rightarrow 0$ was so viel heißen würde, als dass der Wert der Zelle zum Zeitpunkt $t + 1$ unabhängig von dem Wert der links benachbarten Zelle einen Zeitschritt zuvor ist, wenn die Zelle selbst und ihr rechter Nachbar den Wert 1 haben.

Welches physikalische Verhalten könnte man mit diesem Automaten beschreiben?

- e) Für diesen letzten Aufgabenteil müssen Ihnen die Ergebnisse der Programmieraufgabe vorliegen. Teilen Sie die in der Figure gezeigten Regeln in die von Wolfram definierten Klassen i) „homogeneous state“, ii) „separated simple stable or periodic structures“, iii) „chaotic pattern“ iv) „complex localized structures, sometimes long-lived“ ein.

Hinweis zur Abgabe der Theorieaufgabe:

Bitte in schriftlicher Form am 09.11.2017 bei Herrn Dr.-Ing. Markus Heß direkt vor/nach der Vorlesung abgeben.

Programmieraufgabe: Totalistische Regeln

Schreiben Sie eine Funktion `NumSimHA1(n)`, die eine Figure mit 32 Subplots, angeordnet als 4×8 -Grid, ausgibt.

Jeder Subplot soll dabei aus $n \times n$ Kästchen bestehen. Ein Kästchen kann den Zustand 0 (weiß) oder 1 (schwarz) annehmen.

Die oberste Reihe aus n Kästchen eines Subplots soll eine zufällige Binärverteilung sein (weiß oder schwarz mit der Wahrscheinlichkeit 0.5).

Diese Anfangskonfiguration eines Subplots ist eine der n Reihen, welche nach unten hin die zeitliche Entwicklung mit n Schritten zeigen sollen.

Die Update-Regel jedes Subplots sei totalistisch mit dem Nachbarschaftsradius $r = 2$. Die Randbedingungen sollen periodisch gewählt werden.

Jeder Subplot soll eine andere Update-Regel zeigen, beginnend bei der Regel Nr. 0 oben links, und dann reihenweise in zweier Schritten (0, 2, 4, 6, ...) aufsteigend bis zur Regel Nr. 62 im Subplot unten rechts.

Bitte entfernen Sie die Achsen eines jeden Subplots und geben die Nummer der jeweiligen Regel im jeweiligen Titel des Subplots an. Bitte legen Sie für die Ausgabe-Figure keine Größe oder Position fest.

Verwenden Sie für die Erstellung eines einzelnen Subplots bitte nicht mehr als eine Schleife (`for`, `while` ...).

Hinweise zur Abgabe der Programmieraufgabe:

Das Skript `NumSimHA01.m` bitte als Anhang einer E-Mail mit dem Betreff `NumSimHA01` an j.benad@tu-berlin.de senden.

Die Abgabedeadline ist der 09.11.2017 um 12¹⁵ Uhr.

Bitte in dem Skript die folgende Form verwenden:

```
% Nachname1      Matrikelnummer1   (Liste bitte alphabetisch nach Nachnamen ordnen)
% Nachname2      Matrikelnummer2
% Nachname3      Matrikelnummer3
% Nachname4      Matrikelnummer4

function NumSimHA1(n)

    % Hier den Code einfügen. Bitte gut kommentieren.

end
```