Epsilon-Maschinen

und Information Bottlenecks

Malte Harder

Sommerakademie Olang – 17. September 2008

Periodizität und Zufall

- Historische Unterteilung von physikalischen Prozessen in periodische und zufällige.
- Periodische Prozesse sind aufgrund von Wiederholungen „einfach", zufällige Prozesse hingegen besitzen eine einfache statistische Beschreibung.

Periodizität und Zufall

- Historische Unterteilung von physikalischen Prozessen in periodische und zufällige.
- Periodische Prozesse sind aufgrund von Wiederholungen „einfach", zufällige Prozesse hingegen besitzen eine einfache statistische Beschreibung.
- Viele beobachteten Prozesse verhalten sich jedoch chaotisch und liegen „zwischen" den Extremen periodisch und zufällig [3, S. 105].

Modelle

- Ein Agent beobachtet einen Prozess über Sensoren die diskrete Werte mit einem Abstand ε liefern.
- Ziel des Agenten ist es, ein Modell M_{min} seiner Umwelt zu bilden, das sowohl minimal ist, als auch die besten Vorhersagen über den Prozess liefert.

Modelle

- Ein Agent beobachtet einen Prozess über Sensoren die diskrete Werte mit einem Abstand ε liefern.
- Ziel des Agenten ist es, ein Modell M_{min} seiner Umwelt zu bilden, das sowohl minimal ist, als auch die besten Vorhersagen über den Prozess liefert.
- Die Größe dieses Modells wird als ein Maß für Struktur/Komplexität eingeführt: $C_0(x) = ||M_{min}(x|\mathcal{L})||$
- Problem: Sprache $\mathcal{L}$, die das Modell beschreibt, beeinflußt die Größe des Modells [2, S. 1-8].

Komplexität

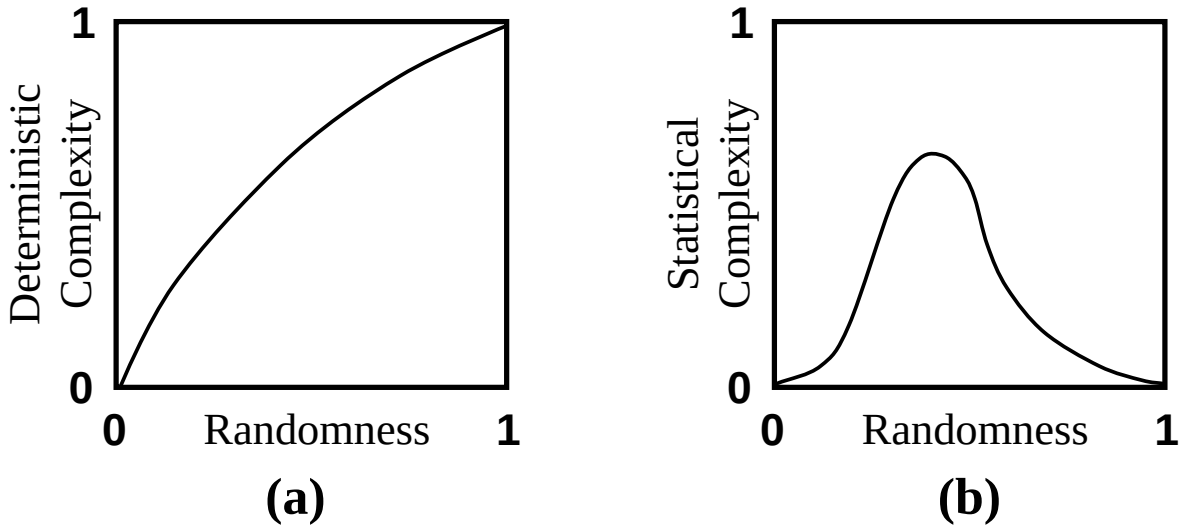

Abbildung: Deterministische vs. statistische Komplexität [2, S. 17]

Entropie & Mutual Information

- *Shannon Entropie* für eine Zufallsvariable X mit der Verteilung $P(X = x) = p_x$ ist definiert als

$$H(P) := H(X) := \sum_{x \in \mathcal{X}} -p_x \log_2 p_x$$

- Bedingte Entropie $H(X|Y) := H(X, Y) - H(Y)$ mit

$$H(X, Y) := \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} -P(X = x, Y = y) \log_2 P(X = x, Y = y)$$

[6]

- Die *Mutual Information* (auch *Transinformation*) zweier Zufallsvariablen ist:

$$I(X; Y) := H(X) - H(X|Y)$$

- Wie ändert sich die Unsicherheit von X wenn Y bekannt ist.
- Ist $I(X; Y) = 0$ dann sind X und Y unabhängig.

Prozesse

- Im folgenden beschränkt auf diskrete bedingt stationäre Prozesse.
- Ein Prozess kann somit als eine unendliche Kette von diskreten Zufallsvariablen S_i, die Werte aus einer abzählbaren Menge $\mathcal{A}$ annehmen, gesehen werden:

$$\overleftrightarrow{S} = ...S_{-1}S_0S_1...$$

Prozesse

- Im folgenden beschränkt auf diskrete bedingt stationäre Prozesse.
- Ein Prozess kann somit als eine unendliche Kette von diskreten Zufallsvariablen S_i, die Werte aus einer abzählbaren Menge $\mathcal{A}$ annehmen, gesehen werden:

$$\overleftrightarrow{S} = ...S_{-1}S_0S_1...$$

- Teilsequenzen werden wie folgt bezeichnet:

$$\overrightarrow{S}_t^L = S_tS_{t+1}...S_{t+L-1}, \overleftarrow{S}_t^L = \overrightarrow{S}_{t-L}^L$$

- Ein Prozess ist bedingt stationär gdw.

$$P(\overrightarrow{S}_t^L = s^L | \overleftarrow{S}_t = \overleftarrow{s}) = P(\overrightarrow{S}_0^L = s^L | \overleftarrow{S}_0 = \overleftarrow{s})$$

für alle $t \in \mathbb{Z}$, $L \in \mathbb{N}_0$ und $s^L \in \mathcal{A}^{\mathcal{L}}$ [4, S. 24].

Effective States

- Eine Partition $\mathfrak{R}$ aller möglichen Vergangenheiten $\overleftarrow{\mathbf{S}}$ wird als *Effective State class* bezeichnet. Ein Element $\rho \in \mathfrak{R}$ als *(effective) State*.
- Ein Prozess befindet sich im Zustand ρ wenn dieser die aktuelle Vergangenheit $\overleftarrow{s}$ enthält.

Effective States

- Eine Partition $\mathfrak{R}$ aller möglichen Vergangenheiten $\overleftarrow{\mathbf{S}}$ wird als *Effective State class* bezeichnet. Ein Element $\rho \in \mathfrak{R}$ als *(effective) State*.
- Ein Prozess befindet sich im Zustand ρ wenn dieser die aktuelle Vergangenheit $\overleftarrow{s}$ enthält.
- Die Funktion $\eta : \overleftarrow{\mathbf{S}} \rightarrow \mathfrak{R}$ ordnet einer Vergangenheit den aktuellen Zustand zu.
- Analog zur Zufallsvariable $\overleftarrow{S}$ der Vergangenheit, gibt es eine Zufallsvariable $\mathcal{R}$ der *Effective States*.
- Für jeden *Effective State* lässt sich eine Verteilung der Zukunft $\overrightarrow{S}$ definieren.[4, S. 25]

Statistische Komplexität

- Die statistische Komplexität einer *Effective State Class* $\mathfrak{R}$ ist über die Entropie der dazugehörigen Zufallsvariable definiert:

$$C_\mu(\mathfrak{R}) = H(\mathcal{R})$$

- C_μ kann als die Anzahl der Bits interpretiert werden, die der Prozess über seinen bisherige Vergangenheit bezüglich der Partition durch $\mathfrak{R}$ speichert.

Das Old Country Lemma

Lemma

Für alle $\mathfrak{R}$ und $L \in \mathbb{N}_0$ gilt,

$$H(\overrightarrow{S}^L|\mathcal{R}) \geq H(\overrightarrow{S}^L|\overleftarrow{S}).$$

Proof.

Nach Konstruktion ist

$$H(\overrightarrow{S}^L|\mathcal{R}) = H(\overrightarrow{S}^L|\eta(\overleftarrow{S})).$$

Es gilt aber

$$H(\overrightarrow{S}^L|\eta(\overleftarrow{S})) \geq H(\overrightarrow{S}^L|\overleftarrow{S}).$$

[4, S. 25,27] □

Causal States

- Die *Causal States* $\mathfrak{S} = \{\mathcal{S}_i | i \in I\}$ eines Prozesses sind definiert als die Äquivalenzklassen unter der Relation

$$\overleftarrow{s} \sim \overleftarrow{s}' \iff \forall \overrightarrow{S} : P(\overrightarrow{S} | \overleftarrow{S} = \overleftarrow{s}) = P(\overrightarrow{S} | \overleftarrow{S} = \overleftarrow{s}')$$

  und bilden somit eine *Effective State Class*.
- Die Zufallsvariable der *Causal States* wird mit $\mathcal{S}$ bezeichnet.

Causal States

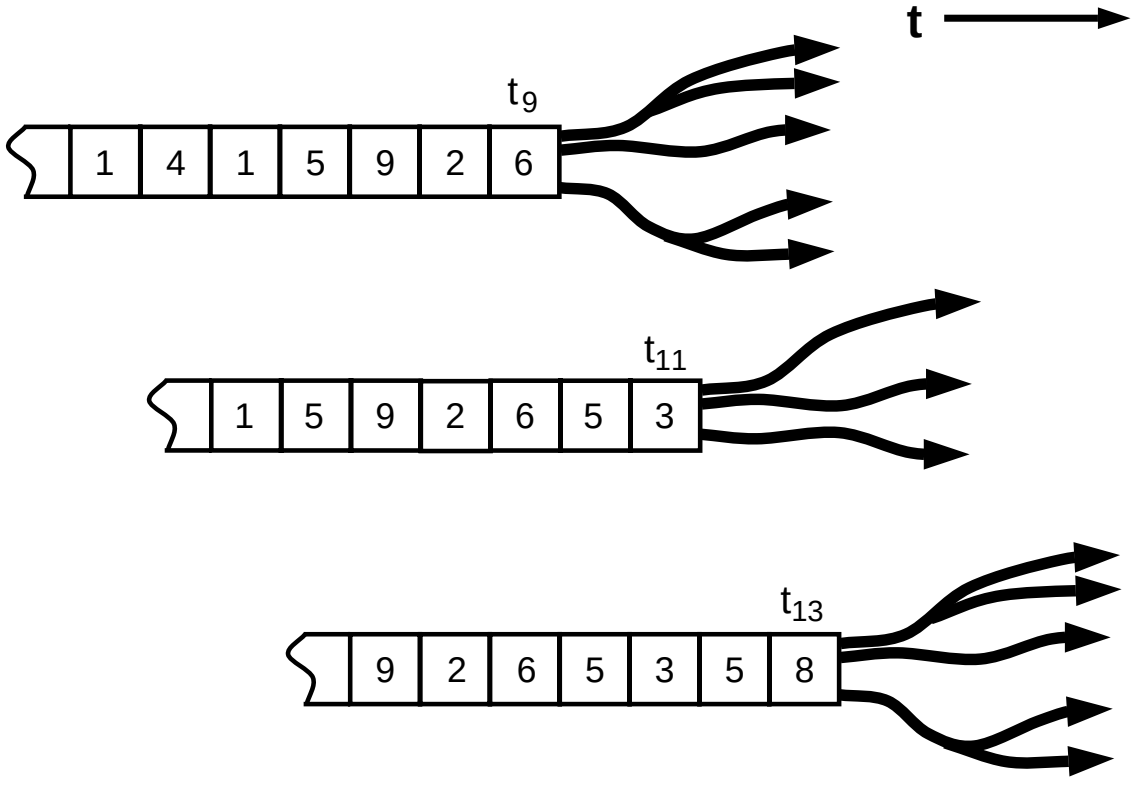

Abbildung: Morphs von *Causal States* [2, S. 18]

Minimalität und Vorhersagen

- *Causal States* ermöglichen die sichersten Vorhersagen bezüglich $\overrightarrow{S}$, denn

  $$H(\overrightarrow{S}^L|\mathcal{S}) = H(\overrightarrow{S}^L|\overleftarrow{S}) \text{ und folglich } H(\overrightarrow{S}^L|\mathcal{R}) \geq H(\overrightarrow{S}^L|\overleftarrow{S}).$$

  Anders ausgedrückt

  $$I(\overrightarrow{S}^L;\mathcal{S}) = I(\overrightarrow{S}^L;\overleftarrow{S}).$$

Minimalität und Vorhersagen

- *Causal States* ermöglichen die sichersten Vorhersagen bezüglich $\overrightarrow{S}$, denn

  $$H(\overrightarrow{S}^L|\mathcal{S}) = H(\overrightarrow{S}^L|\overleftarrow{S}) \text{ und folglich } H(\overrightarrow{S}^L|\mathcal{R}) \geq H(\overrightarrow{S}^L|\overleftarrow{S}).$$

  Anders ausgedrückt

  $$I(\overrightarrow{S}^L;\mathcal{S}) = I(\overrightarrow{S}^L;\overleftarrow{S}).$$

- Außerdem sind sie minimal bezüglich C_μ,

  $$C_\mu(\mathcal{R}) \geq C_\mu(\mathcal{S})$$

  daher definiert man nun die statistische Komplexität eines Prozesses $C_\mu(O)$ über die seiner *Causal States*:

  $$C_\mu(O) := C_\mu(\mathcal{S})$$

- *Causal States* sind eindeutig bestimmt.

Prädiktive Information

- Prädiktive Information (auch *Excess Entropy*) ist die *Mutual Information* zwischen Vergangenheit und Zukunft eines Prozesses:

$$\mathbf{E} := I(\overrightarrow{S}; \overleftarrow{S})$$

- **E** misst dabei die „sichtbar" gespeicherte Information des Prozesses über die Vergangenheit.

Prädiktive Information

- Prädiktive Information (auch *Excess Entropy*) ist die *Mutual Information* zwischen Vergangenheit und Zukunft eines Prozesses:

$$\mathbf{E} := I(\overrightarrow{S}; \overleftarrow{S})$$

- **E** misst dabei die „sichtbar" gespeicherte Information des Prozesses über die Vergangenheit.
- Die wirklich gespeicherte Information des Prozesses ist jedoch C_μ und es gilt $\mathbf{E} \leq C_\mu$.

Epsilon-Maschinen

- Die Wahrscheinlichkeit des Wechsels von Zustand $\mathcal{S}_i$ zu $\mathcal{S}_j$ bei Auftreten des Symbols $s \in \mathcal{A}$ wird mit $T_{ij}^{(s)}$ bezeichnet
- Die Menge $\mathbf{T} := \{T_{ij}^{(s)} | s \in \mathcal{A}, i, j \in I\}$ bildet zusammen mit der Funktion ε die Epsilon-Maschine $\{\varepsilon, \mathbf{T}\}$ des Prozesses.
- Epsilon-Maschinen sind deterministisch und Markovsch: Der Nachfolgezustand wird durch ein weiteres Zeichen eindeutig bestimmt und hängt nur vom aktuellen Zustand ab. Eine Epsilon Maschine lässt sich also als ein *Deterministic Finite Automaton* darstellen. [4, S. 27-32]
- Epsilon-Maschinen bilden ein Monoid, dessen Untergruppen die Symmetrien des Prozesses darstellen. [4, S. 133]

Epsilon-Maschinen

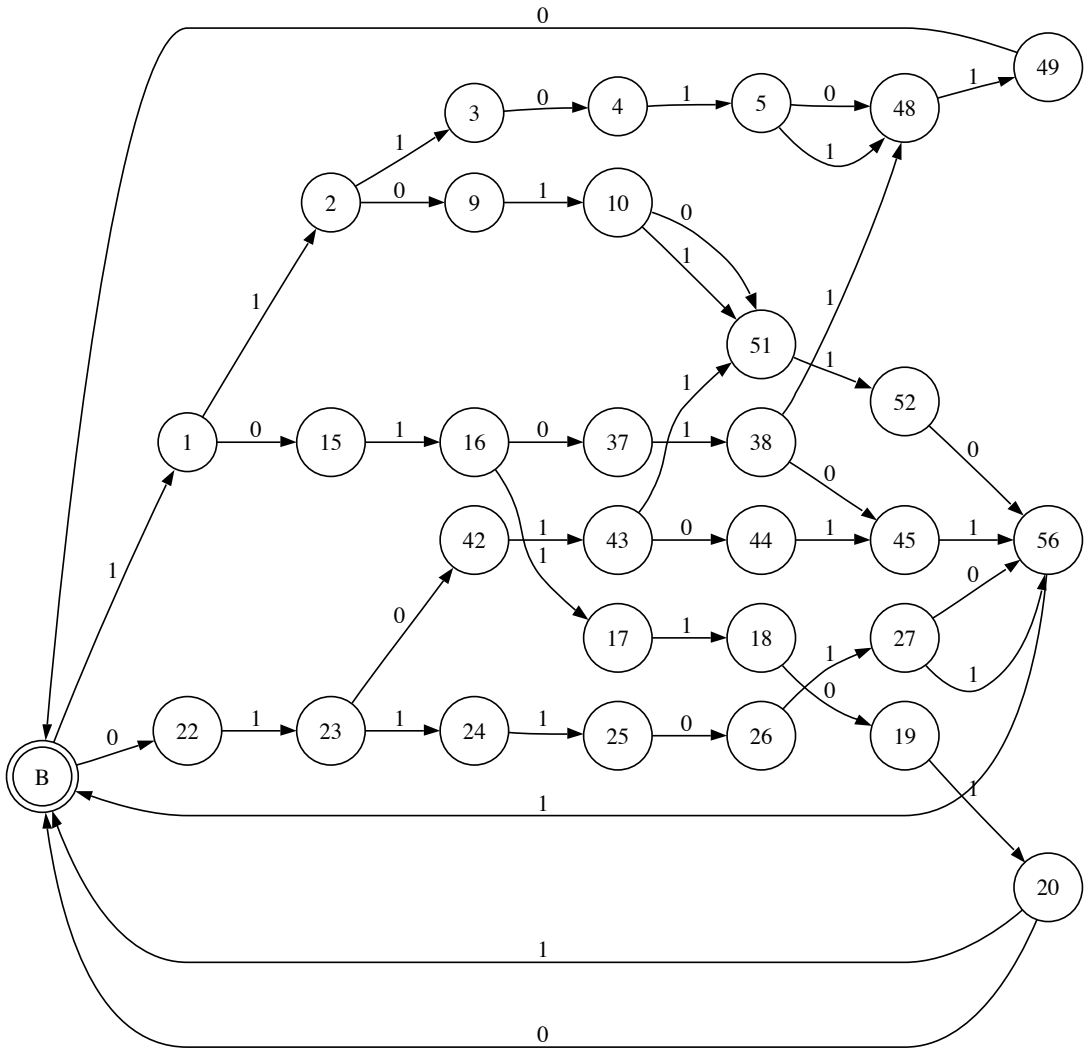

Abbildung: Epsilon Maschine für die logistische Abbildung mit $r = 3.57$

Bottleneck-Variable

- Problemstellung: Minimierung des Funktionals

$$\mathcal{L}[p(\tilde{x}|x)] = I(\tilde{X};X) - \beta I(\tilde{X};Y) \text{ [1, S. 6]}.$$

- Für den Fall $\beta \to \infty$ mit $X = \overleftarrow{S}$ und $Y = \overrightarrow{S}$ ergibt $\tilde{X}$ genau $\mathcal{S}$, d.h. die Zufallsvariable der *Causal States*.

Bottleneck-Variable

- Problemstellung: Minimierung des Funktionals

$$\mathcal{L}[p(\tilde{x}|x)] = I(\tilde{X};X) - \beta I(\tilde{X};Y) \text{ [1, S. 6].}$$

- Für den Fall $\beta \to \infty$ mit $X = \overleftarrow{S}$ und $Y = \overrightarrow{S}$ ergibt $\tilde{X}$ genau $\mathcal{S}$, d.h. die Zufallsvariable der *Causal States*.
- Für $\beta \to \infty$ gilt es $I(\tilde{X};Y)$ zu maximieren, damit $\mathcal{L}$ minimal wird.
- Es folgt, wegen $H(\overrightarrow{S}|\mathcal{R}) \geq H(\overrightarrow{S},\mathcal{S})$, dass das Funktional mit $\tilde{X} = \mathcal{S}$ minimal wird [5, S. 2].

Rekonstruktion

- Teilweise ist es möglich für einen gegebenen (oder die Approximation eines) Prozess die Epsilon-Maschine analytisch zu berechnen.
- Auch aus empirischen Daten können über Algorithmen Epsilon-Maschinen rekonstruiert und damit die *Causal States* des Prozesses bestimmt werden.
- Ein Beispiel dafür ist der Algorithmus nach Crutchfield und Young [3, S. 106]. Dieser ist einfach zu implementieren, liefert aber aufgrund von Approximation teilweise ungewünschte Ergebnisse zurück.

Hierarchische Rekonstruktion

- Eingangs erwähnt:
  - Ziel des Agenten ist es, ein Modell M_{min} seiner Umwelt zu bilden, das sowohl minimal ist (Occam's principle), als auch die besten Vorhersagen über den Prozess liefert.
  - Die Größe dieses Modells wird als ein Maß für Struktur/Komplexität eingeführt: $C_0(x) = ||M_{min}(x|\mathcal{L})||$

Hierarchische Rekonstruktion

- Eingangs erwähnt:
  - Ziel des Agenten ist es, ein Modell M_{min} seiner Umwelt zu bilden, das sowohl minimal ist (Occam's principle), als auch die besten Vorhersagen über den Prozess liefert.
  - Die Größe dieses Modells wird als ein Maß für Struktur/Komplexität eingeführt: $C_0(x) = ||M_{min}(x|\mathcal{L})||$
- Idee: Divergiert der Speicheraufwand des Modelles bei kleinem ε, wähle mächtigere Modellklasse z.B. von *DFA* zu *Stack Automata*.
- Damit verallgemeinert sich die Definition der Epsilon-Maschine zum kleinsten Modell der am wenigsten mächtigen Modellklasse, dass eine endliche Beschreibung liefert.
- Problem: Es gibt keine natürliche Hierarchie von Modellklassen. Außerdem besitzt ein Agent keine Möglichkeit Innovation zu betreiben. [3, S. 23 ff]

Logistische Abbildung

- Die logistische Abbildung ist definiert als

$$x_{n+1} = f(x_n) \text{ mit } f(x) = rx(1-x), x_0 \in [0,1]$$

- Je nach Wahl des Parameters $r \in [0,4]$ kann man verschiedene Verhaltensweisen der Orbits $x_0 x_1 x_2 \ldots$ beobachten.

Logistische Abbildung

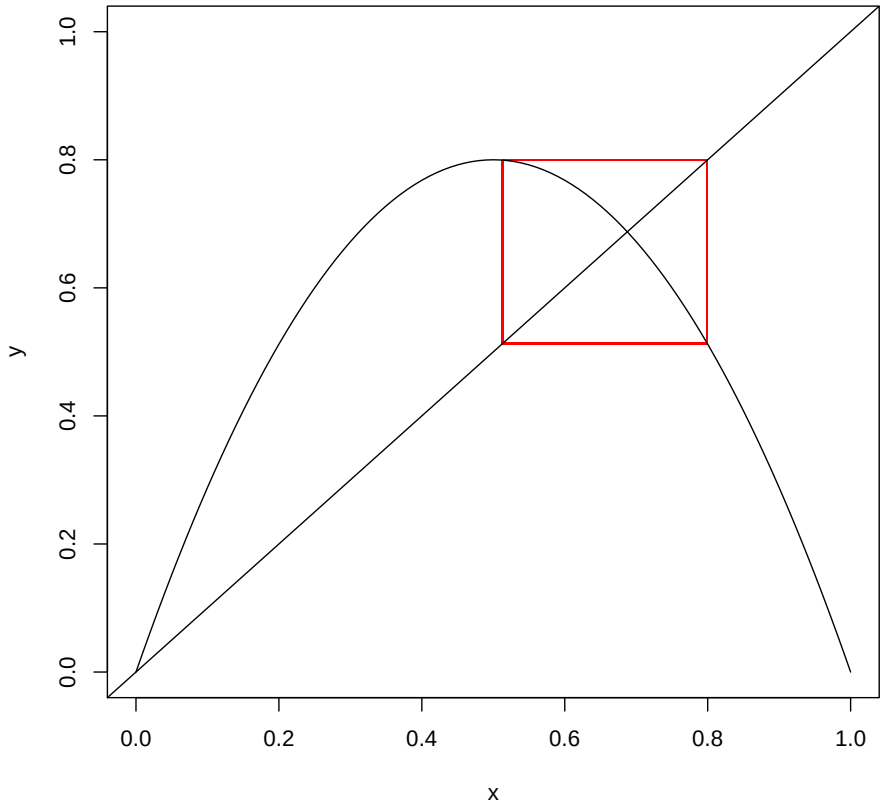

Abbildung: 2-er Periode für $r = 3.2$

Logistische Abbildung

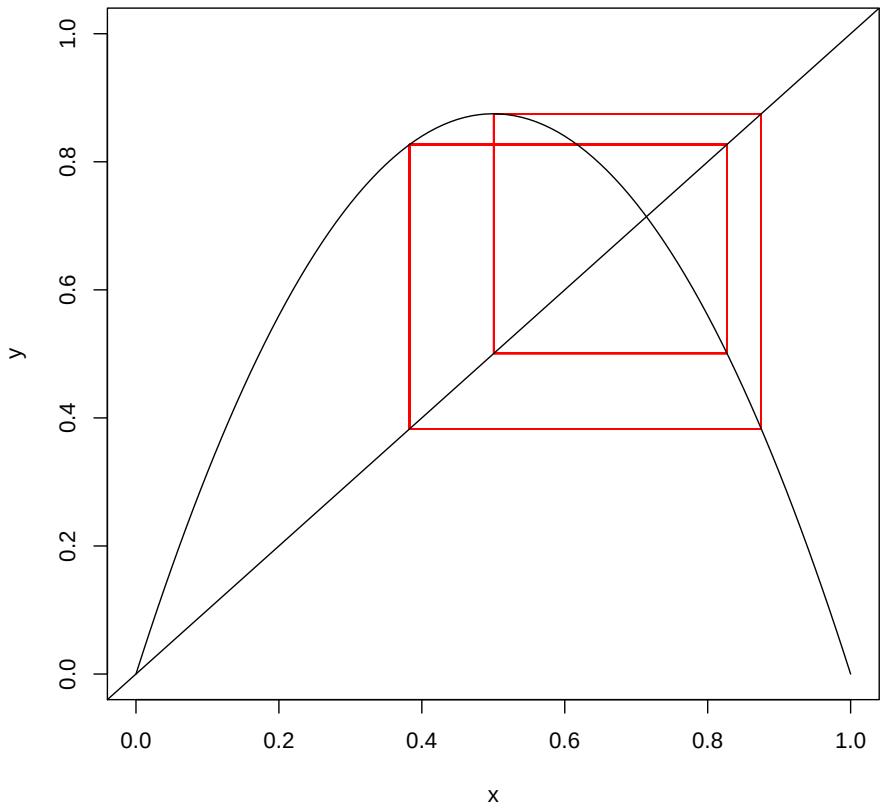

Abbildung: 4-er Periode für $r = 3.5$

Logistische Abbildung

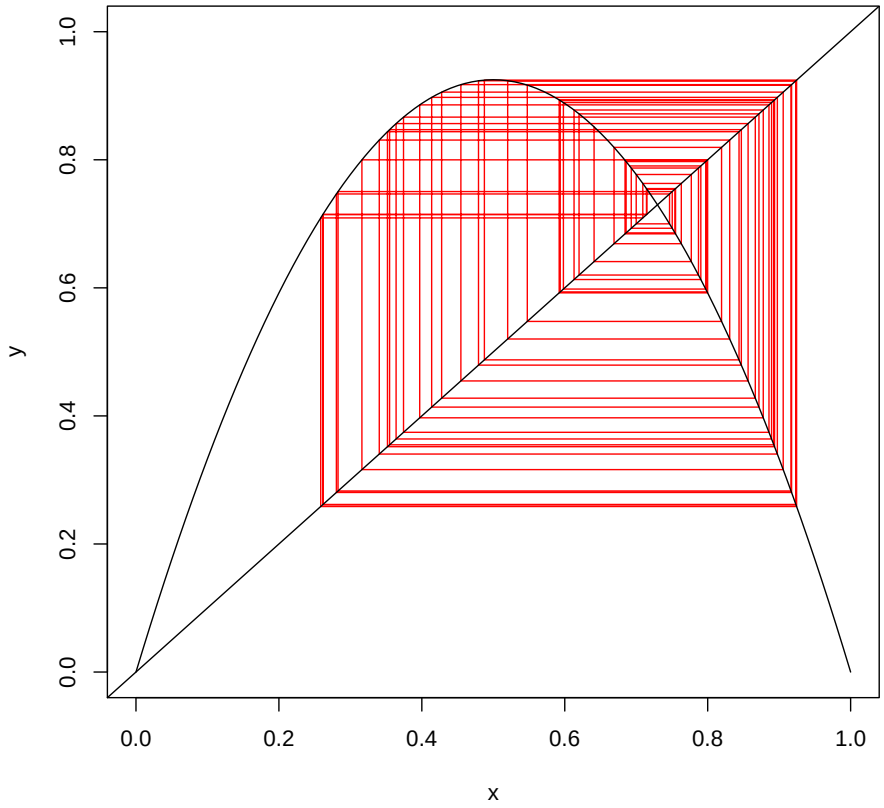

Abbildung: Chaotisches Verhalten für $r = 3.7$

Logistische Abbildung

- Die logistische Abbildung ist definiert als

$$x_{n+1} = f(x_n) \text{ mit } f(x) = rx(1-x), x_0 \in [0,1]$$

- Je nach Wahl des Parameters $r \in [0,4]$ kann man verschiedene Verhaltensweisen der Orbits $x_0 x_1 x_2 \ldots$ beobachten.
- Kodierung eines Orbits durch binäre Sequenz: 0 für $x_i < \frac{1}{2}$, 1 für $x_i \geq \frac{1}{2}$.
- Diese Sequenz ist generierend, das heißt hinreichend lange Sequenzen beschreiben beliebig kleine Umgebungen von x_0.
- Idee: Komplexität von f über die Epsilon-Maschinen der binären Sequenz beschreiben.

Logistische Abbildung

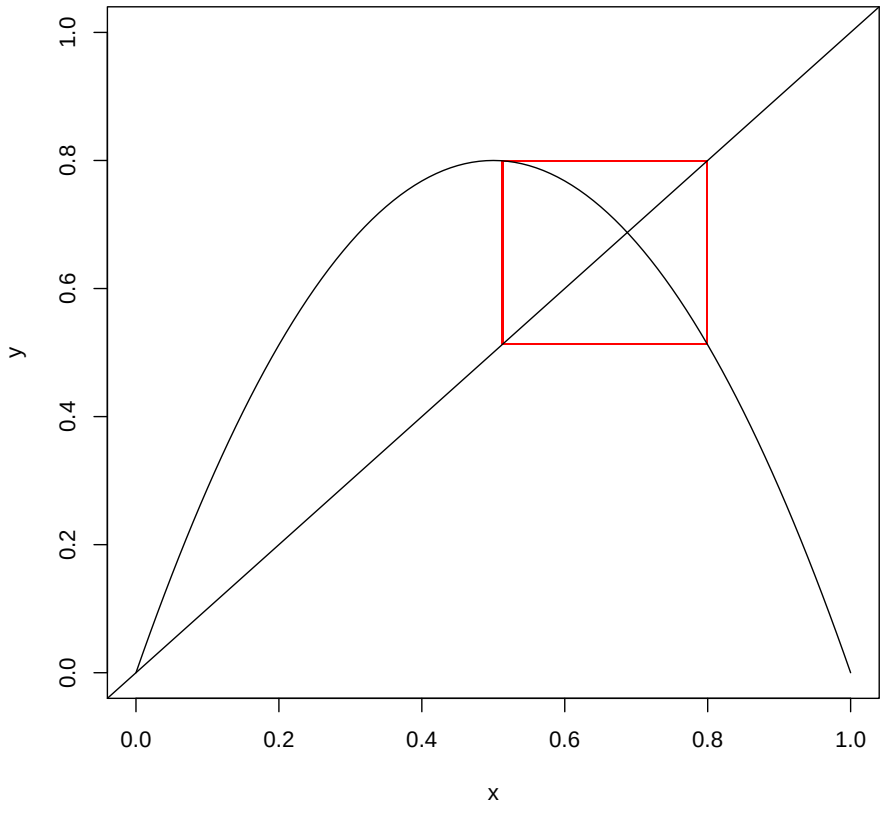

Abbildung: 2-er Periode für $r = 3.2$

Binäre Sequenz: 11111111111111111111111111111111111111111111111111111111111111111111

Logistische Abbildung

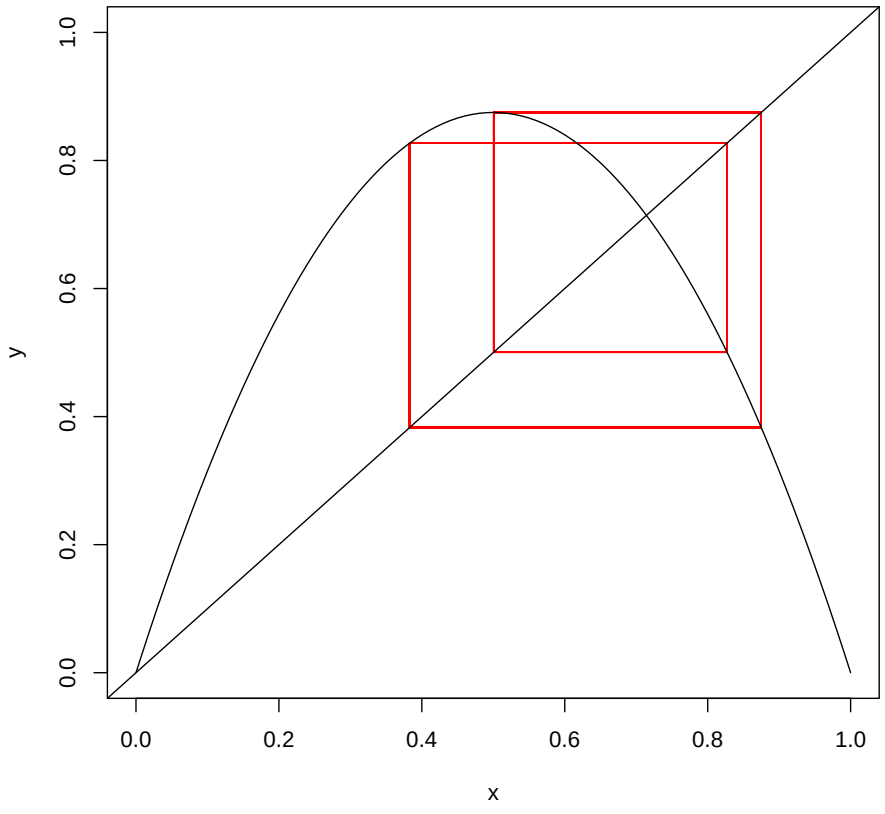

Abbildung: 4-er Periode für $r = 3.5$

Binäre Sequenz: 11101110111011101110111011101110111011101110111011101110111011101110

Logistische Abbildung

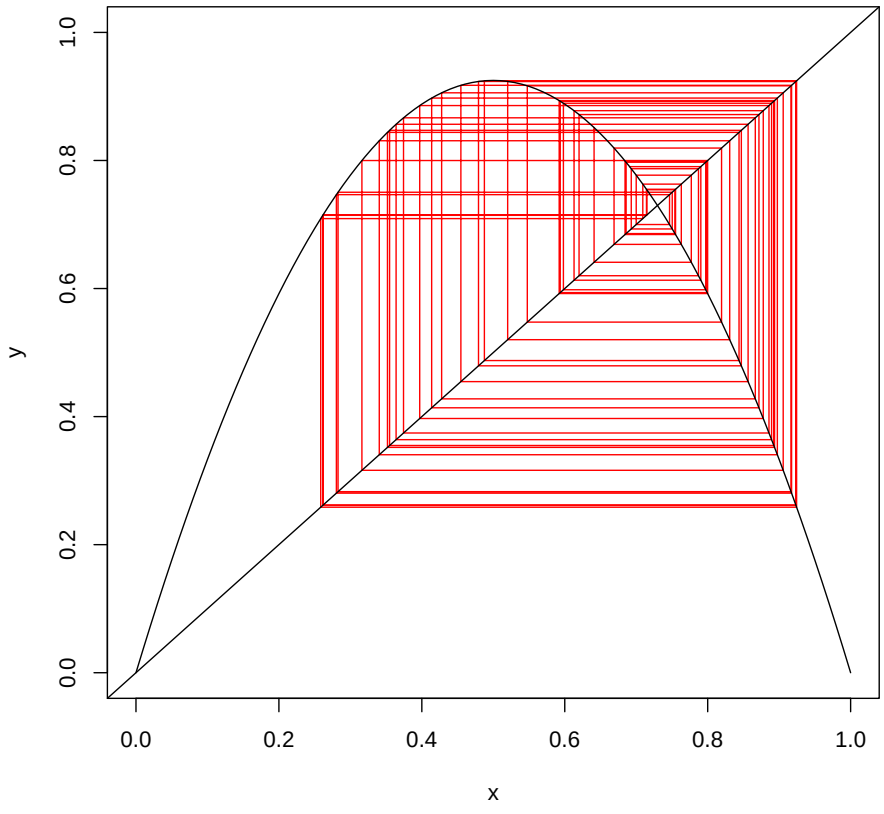

Abbildung: Chaotisches Verhalten für $r = 3.7$

Binäre Sequenz: 1111010101011101010111111011111010111011111101010111111010101111

Rekonstruktionsalgorithmus von Crutchfield

1. Für eine Sequenz des Prozesses der Länge n, bilde einen Baum aus allen Teilsequenzen der Länge L.

Rekonstruktionsalgorithmus von Crutchfield

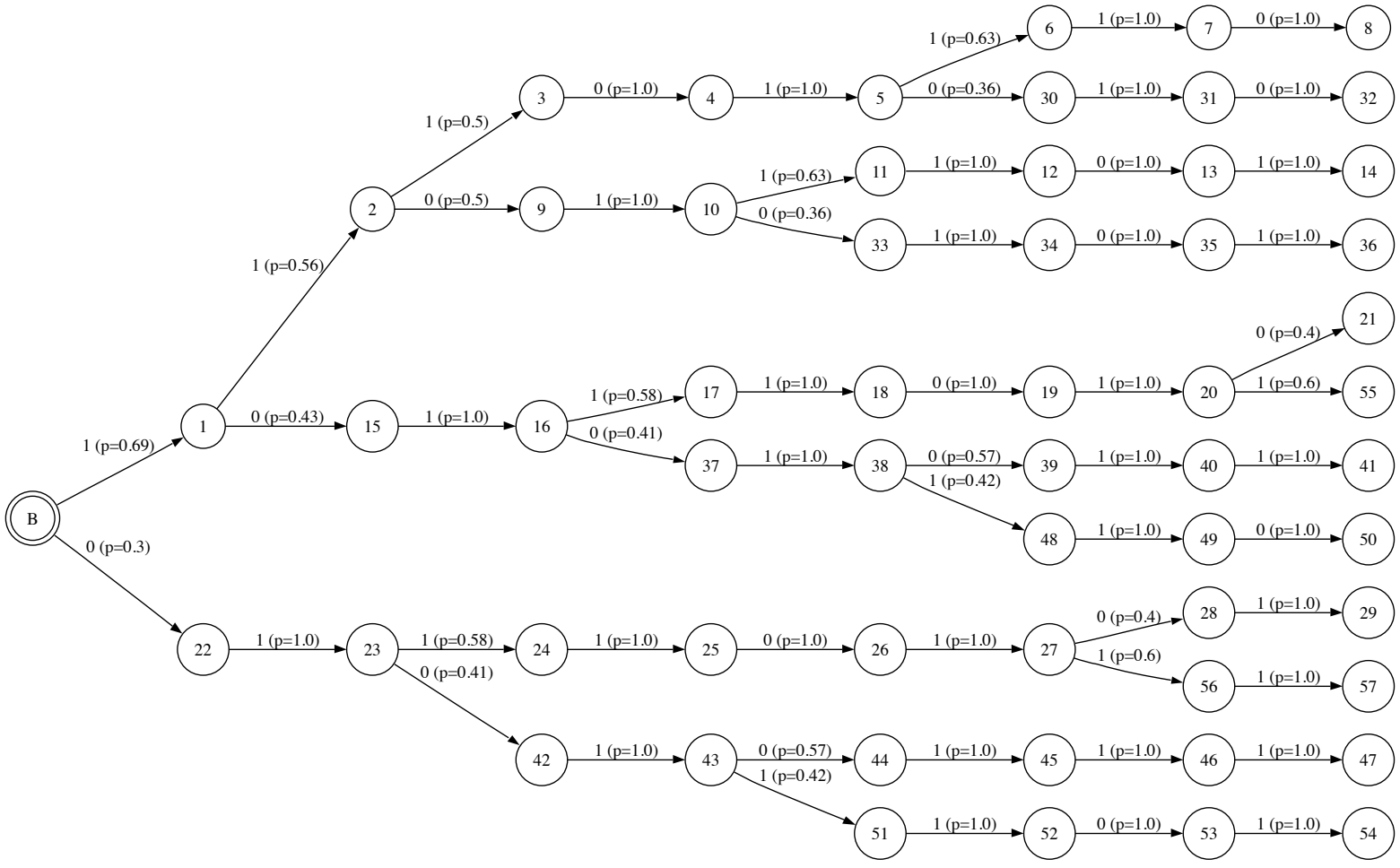

Abbildung: Baum für $r = 3.569$

Rekonstruktionsalgorithmus von Crutchfield

1. Für eine Sequenz des Prozesses der Länge n, bilde einen Baum aus allen Teilsequenzen der Länge L.
2. Bezeichne alle Nodes, deren Unterbaum**struktur** bis zur Tiefe $l \leq L$ äquivalent ist, als Äquivalent.

Rekonstruktionsalgorithmus von Crutchfield

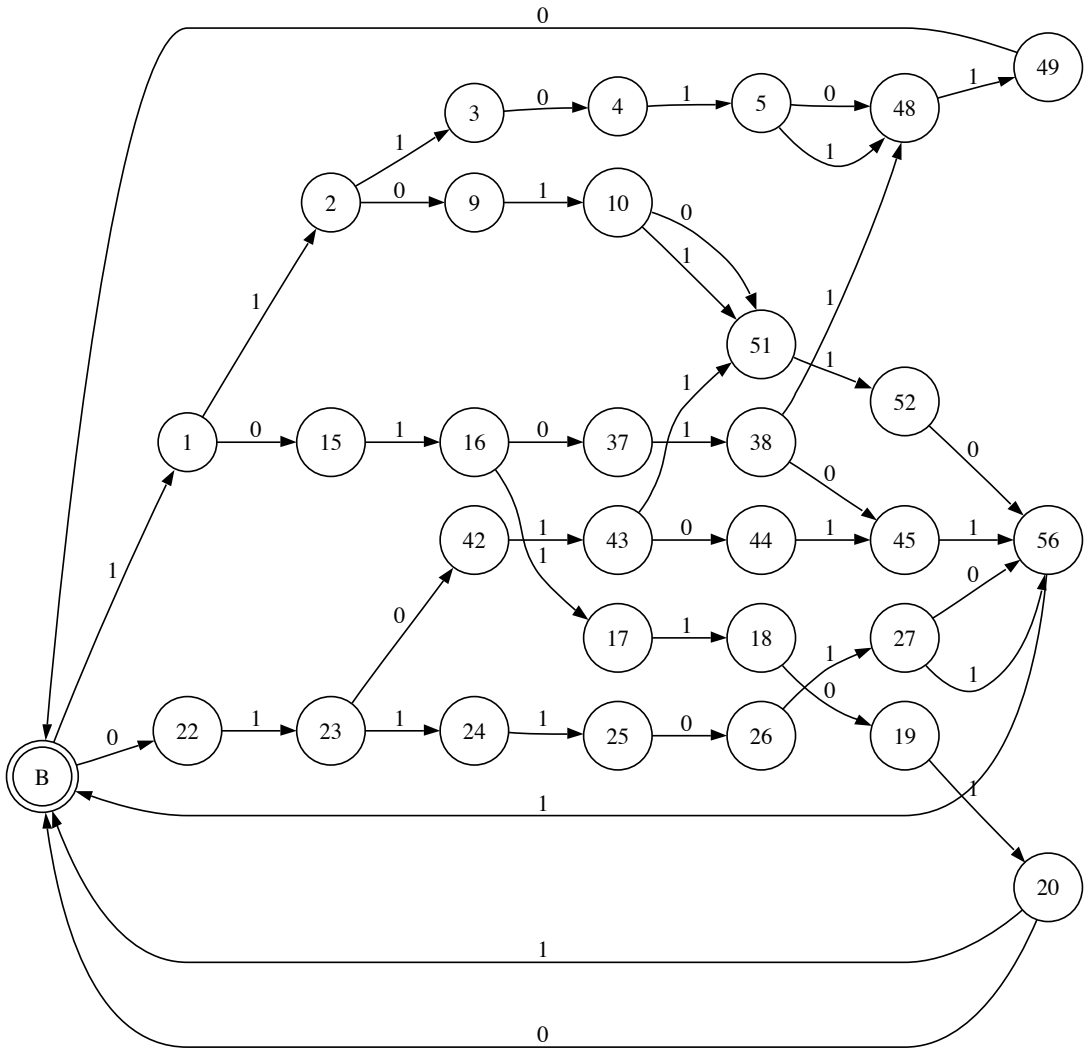

Abbildung: Epsilon-Maschine für $r = 3.569$

Rekonstruktionsalgorithmus von Crutchfield

1. Für eine Sequenz des Prozesses der Länge n, bilde einen Baum aus allen Teilsequenzen der Länge L.
2. Bezeichne alle Nodes, deren Unterbaum**struktur** bis zur Tiefe $l \leq L$ äquivalent ist, als Äquivalent.
   - Der Baum wird dadurch zu einem Graphen.
   - Nimmt man neben der Struktur als Äquivalenzkriterium auch die Wahrscheinlichkeit einer Edge, dann erhält man eine metrische Epsilonmaschine.

Rekonstruktionsalgorithmus von Crutchfield

1. Für eine Sequenz des Prozesses der Länge n, bilde einen Baum aus allen Teilsequenzen der Länge L.
2. Bezeichne alle Nodes, deren Unterbaum**struktur** bis zur Tiefe $l \leq L$ äquivalent ist, als Äquivalent.
   - Der Baum wird dadurch zu einem Graphen.
   - Nimmt man neben der Struktur als Äquivalenzkriterium auch die Wahrscheinlichkeit einer Edge, dann erhält man eine metrische Epsilonmaschine.

- Der Logarithmus der Anzahl der Nodes der entstehenden Maschine wird mit C_0 bezeichnet, für den metrischen Fall lässt sich $C_\mu = H(\mathcal{S})$ auch aus dem Graphen berechnen.
- Algorithmus kann für bestimmte Kombinationen von L, l und ε indeterministische Automaten zurückgeben, dann heißt die Maschine für dieses Parameterset nicht rekonstruierbar [3, S. 105 ff].

Epsilonmaschinen für variierte Parameter von r

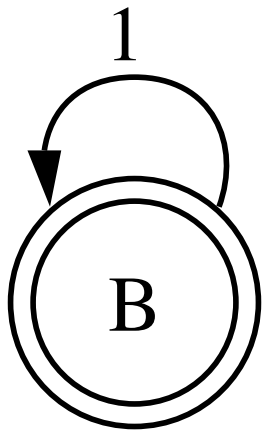

Abbildung: Epsilon-Maschine für $r = 3.2$, $H = 0$, $C_0 = 0$

Epsilonmaschinen für variierte Parameter von r

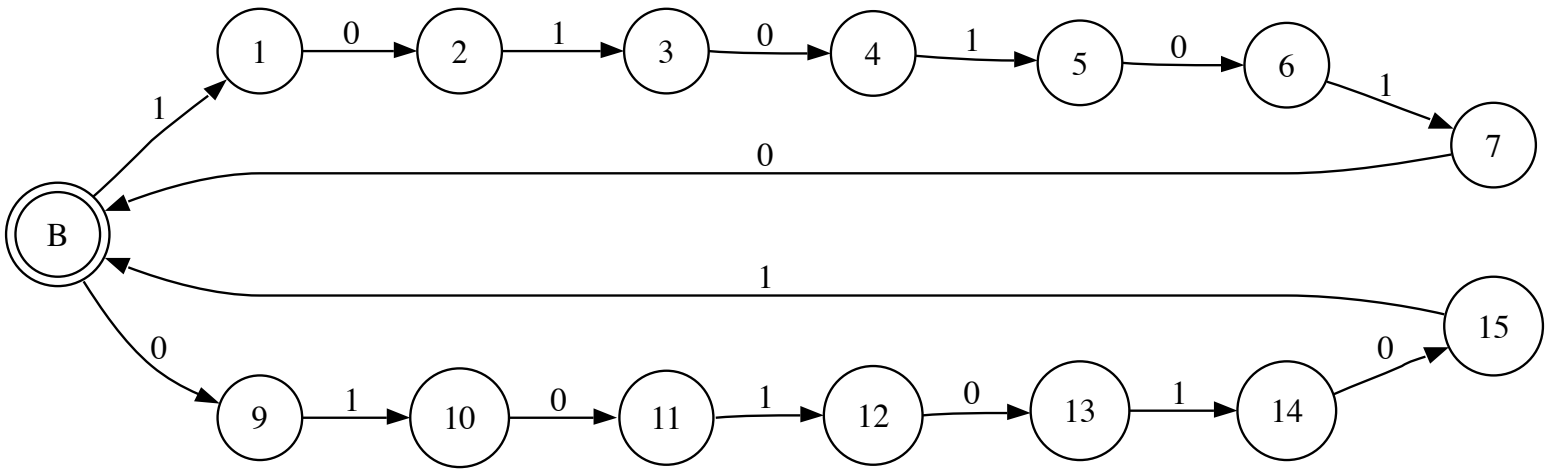

Abbildung: Epsilon-Maschine für $r = 3.45$, $H \approx 0.5$, $C_0 \approx 3.90$

Epsilonmaschinen für variierte Parameter von r

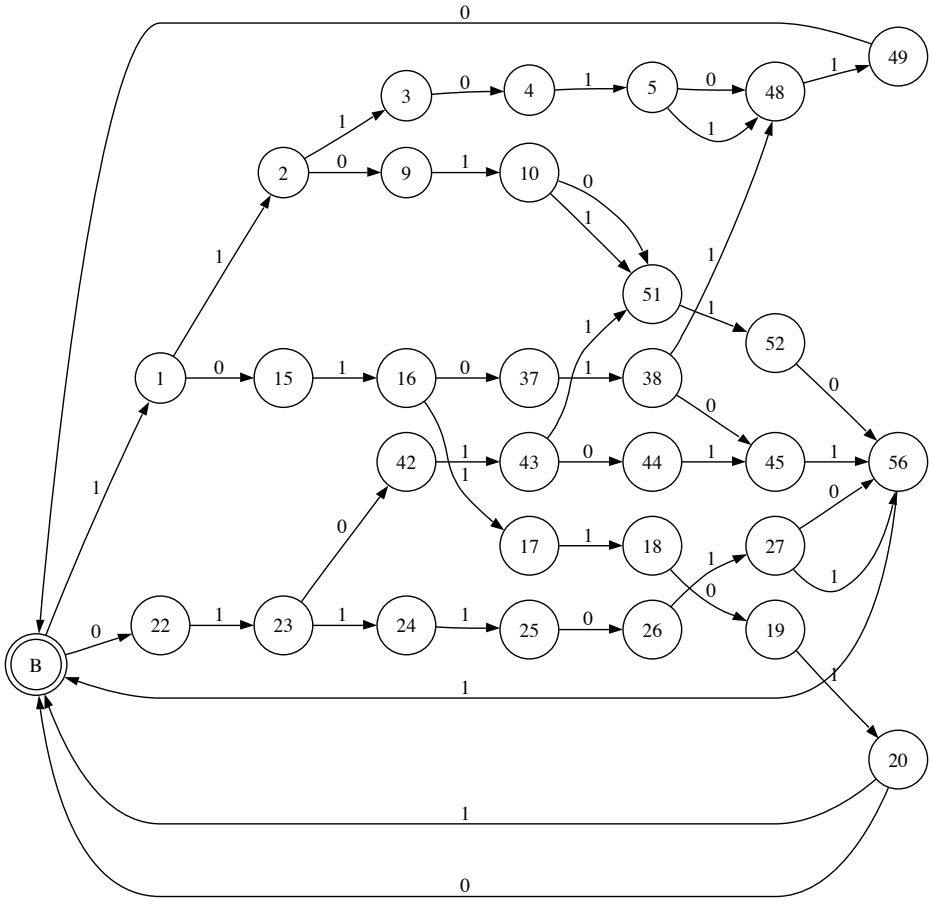

Abbildung: Epsilon-Maschine für $r = 3.569$, $H \approx 0.88$, $C_0 \approx 4.95$

Epsilonmaschinen für variierte Parameter von r

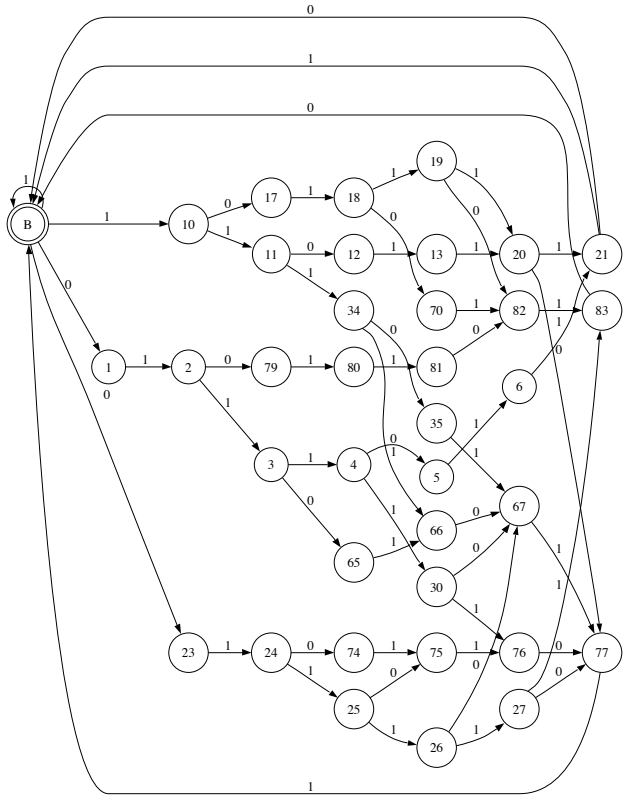

Abbildung: Epsilon-Maschine für $r = 3.74$, $H \approx 0.80$, $C_0 \approx 5.20$

Epsilonmaschinen für variierte Parameter von r

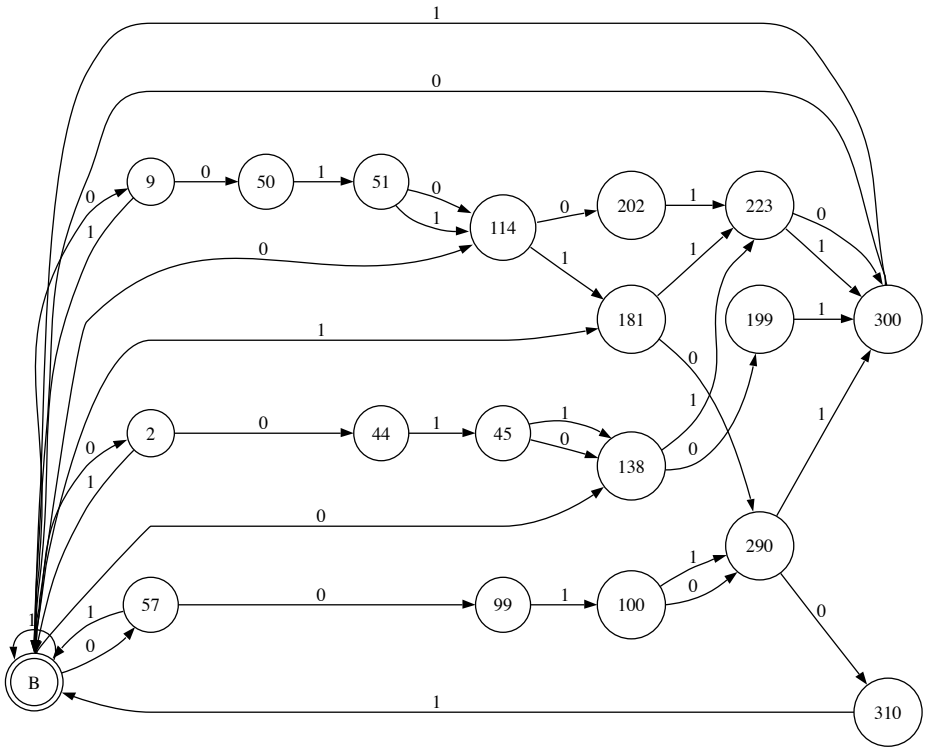

Abbildung: Epsilon-Maschine für $r = 3.95$, $H \approx 0.97$, $C_0 \approx 4.24$

Epsilonmaschinen für variierte Parameter von r

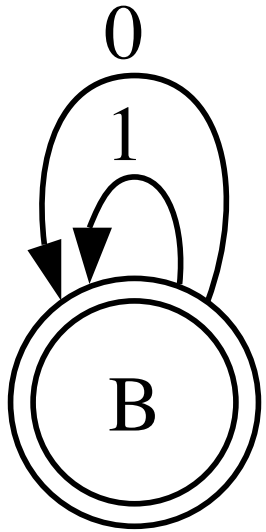

Abbildung: Epsilon-Maschine für $r = 4$, $H \approx 0.99$, $C_0 = 0$

Bibliographie I

W. Bialek, I. Nemenman, and N. Tishby.

Predictability, complexity, and learning.

J. Crutchfield.

The calculi of emergence: computation, dynamics and induction.

Physica D, (Special Issue on the Proceedings of the Oji International Seminar), Jan 1993.

J. Crutchfield and K. Young.

Inferring statistical complexity.

Physical Review Letters, 63(July):105–108, Jan 1989.

C. Shalizi.

Causal architecture, complexity and self-organization in time series and cellular automata.

Unpublished doctoral dissertation, Jan 2001.

Bibliographie II

C. Shalizi and J. Crutchfield.

Information bottlenecks, causal states, and statistical relevance bases: How to represent relevant ....

Arxiv preprint nlin.AO, Jan 2000.

C. E. Shannon.

A mathematical theory of communication.

Bell System Technical Journal, 27(October):379–423, Sep 1948.

Vielen Dank für Eure Aufmerksamkeit!