Biochemisch inspirierte Berechnungen

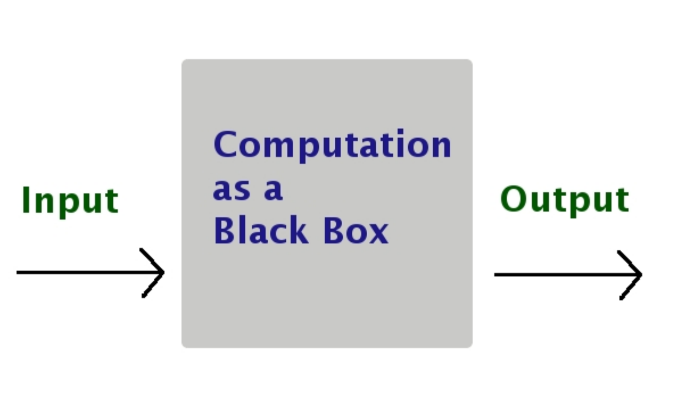
Beobachtersysteme

Peter Leupold

Vorlesung Wintersemester 2009/2010

Motivation für Berechnen durch Beobachtung

Das Standardmodell der Informationsverarbeitung in der Informatik:



wobei Eingabe und Ausgabe in der Regel dasselbe Format haben.

Motivation für Berechnen durch Beobachtung

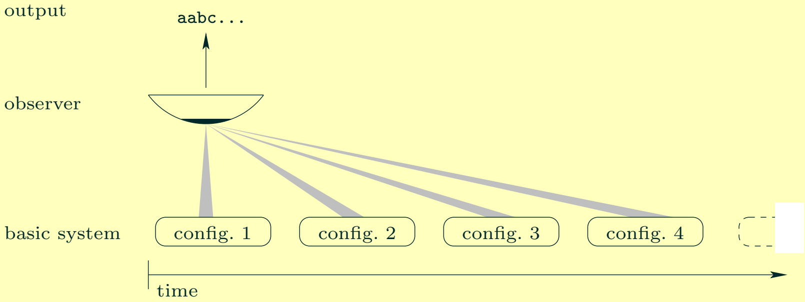
Die meisten Modelle des Natural Computing folgen dieser Standardarchitektur.

Lediglich die Mechanismen in der Black Box orientieren sich an einem natürlichen Prozess.

Biologen und Chemiker gehen aber häufig ganz anders an ihre Experimente heran.

Zum Beispiel bei der Räuber-Beute-Beziehung oder einer katalysierten Reaktion lässt sich das Wesentliche nur im dynamischen Verhalten des Systems erkennen, nicht in einem Endprodukt.

Grundlegende Struktur eines Beobachtersystems



Beobachtersysteme – Defintion

Definition

Ein *erzeugendes Beobachtersystem*, kurz G/O-System, ist ein Tripel $\Omega := [R, s, \mathcal{O}]$ wobei $[\Sigma, R]$ ein Textersetzungssytem über dem Alphabet Σ ist, das aus allen in der Regelmenge R vorkommenden Buchstaben besteht, und $\mathcal{O}$ ist ein monadischer Transducer mit Eingabealphabet Σ. $s \in \Sigma$ ist das Startsymbol.

Die erzeugte Sprache ist:

$$L(\Omega) := \{\mathcal{O}(w_1) \cdot \mathcal{O}(w_2) \cdots \mathcal{O}(w) : s \Rightarrow_R w_1 \Rightarrow_R w_2 \Rightarrow_R \cdots \Rightarrow_R w\}$$

wobei $w \in IRR(R)$ nicht weiter reduzierbar ist.

Beobachtersysteme – Ein Beispiel

Beispiel

Sei $[R, s, \mathcal{O}]$ ein G/O-System, mit der kontextfreien Regelmenge $R = \{s \to rt, s \to arb, r \to arb, r \to ab, t \to bta, t \to ba\}$.

Der Beobachter berechnet folgende Funktion

$$\mathcal{O}(w) = \begin{cases} a & \text{wenn } w \text{ sowohl } r \text{ als auch } t \text{ enthält }, \\ b & \text{wenn } w \text{ ein } r \text{ enthält, aber kein } t, \\ c & \text{wenn } w \text{ weder } r \text{ noch } t \text{ enthält }, \\ \perp & \text{sonst.} \end{cases}$$

Die vier Klauseln sind durch reguläre Ausdrücke mit disjunkten Sprachen beschreibbar.

Beobachtersysteme – Varianten

Definition

$L_{x,y}(R, s, \mathcal{O})$ oder $L_{x,y}(\Omega)$ ist die Sprache, die von einem Beobachtersystem $\Omega = [R, s, \mathcal{O}]$ erzeugt wird

- wobei $x = \perp$ heisst, dass der Beobachter Berechnungen zurückweisen kann, anderenfalls ist $x = \lambda$, und
- wenn $y = a$ kann der Beobachter keine leere Ausgabe produzieren, wenn $y = f$ dann kann der Beobachter frei zwischen leerer und nichtleerer Ausgabe wechseln,
  wenn $y = i$ dann kann der Beobachter anfangs λ ausgeben, nach der ersten nichtleeren Ausgabe allerdings nicht mehr.

Mit der Schreibweise $\mathcal{L}_{x,y}(\mathcal{C}, \mathcal{A})$ bezeichnen wir die Klasse aller Sprachen, die durch Beobachtersysteme mit Textersetzungssystemen aus der Klasse $\mathcal{C}$ und Beobachter aus der Klasse $\mathcal{A}$ erzeugt werden können.

Stets schreibende Systeme

Stets schreibende Systeme mit kontextfreier Textersetzung können alle kontextfreien Sprachen erzeugen.

Dies lässt sich relativ einfach über die Greibach Normalform zeigen.

Es lassen sich aber auch nicht kontextfreie Sprachen erzeugen.

Stets schreibende Systeme

Beispiel

$$[\{s, A, B, C, t\}, \{s \to A, A \to AB, A \to C, B \to C, C \to t\}].$$

Gute Ableitungen:

$$s \Rightarrow A \overset{n-1}{\Rightarrow} AB^{n-1} \Rightarrow CB^{n-1} \overset{n-1}{\Rightarrow} C^n \Rightarrow tC^{n-1} \overset{n-1}{\Rightarrow} t^n.$$

Der Beobachter

$$\mathcal{O}(w) = \begin{cases} a & \text{wenn } w \in AB^*, \\ b & \text{wenn } w \in C^+B^*, \\ c & \text{wenn } w \in t^+C^*, \\ \perp & \text{sonst.} \end{cases}$$

Die erzeugte Sprache ist $\{a^n b^n c^n \mid n > 0\}$.

Beispiel

Das Textersetzungssystem
$\{S \to A, A \to BB, B \to CC, C \to AA, A \to BT_1, B \to CT_2, C \to AT_3,$
$A \to t, B \to t, C \to t, T_1 \to t, T_2 \to t, T_3 \to t\}$
wird benutzt, um vollständige binäre Bäume zu erzeugen.
Durch Ausgabe von a in jedem Schritt werden z.B. a^2 und a^8 durch

$$S \Rightarrow A \Rightarrow t \text{ und}$$

$$S \Rightarrow A \Rightarrow BB \Rightarrow CCB \Rightarrow CCCT_2 \Rightarrow tCCT_2 \Rightarrow ttCT_2 \Rightarrow tttT_2 \Rightarrow tttt$$

erzeugt.

$$\mathcal{A}(w) = \begin{cases} a & \text{wenn} \quad w \in A^+B^* \cup B^+C^* \cup C^+A^* \cup \\ & \qquad\qquad t^*A^*T_3 \cup t^*B^*T_1 \cup t^*C^*T_2 \cup t^+, \\ \perp & \text{sonst.} \end{cases}$$

Initiale Systeme

Wenn wir die implizite Zeit- und Raumbeschränkung der stets schreibenden Systeme aufheben, dann können wir alle rekursiv aufzählbaren Sprachen erzeugen.

Satz

$\mathcal{L}_{\perp,i}(CF, \mathcal{MT}) = RE.$

Beweis über Simulation einer Grammatik in Kuroda-Normalform.

Korollar

Jede rekursiv aufzählbaren Sprache über einem Alphabet Σ ist ein linker Quotient einer Sprache $L_{\perp,a}(R, s, \mathcal{O})$ mit c^*, wobei $c \notin \Sigma$, R eine Menge kontextfreier Textersetzungsregeln und $\mathcal{O}$ ein monadischer Transducer mit Ausgabealphabet $\Sigma \cup \{c\}$ ist.

Freie Systeme

Wenn dem Beobachter der beliebige Wechsel zwischen leerer und nichtleerer Ausgabe erlaubt ist, dann genügen sogar wesentlich eingeschänktere Textersetungssysteme.

Satz

$\mathcal{L}_{\perp,f}(LCCF, \mathcal{MT}) = RE.$

Beweis über Zwei-Zähler-Automaten mit Transitionen aus

$$Q \times \Sigma \times \{z, nz\} \times \{z, nz\} \times Q \times \{\text{stay}, \text{right}\} \times \{+1, 0, -1\}, \times \{+1, 0, -1\}.$$