Die Q-Funktion

Man definiert die Q-Funktion wie folgt:

$$Q(s,a) \equiv r(s,a) + \gamma V^*(\delta(s,a))$$

Damit läßt sich π^* schreiben als

$$\pi^*(s) = arg\max_a Q(s,a)$$

D.h.: Die optimale Policy kann erlernt werden, indem Q gelernt wird, auch wenn r und δ nicht bekannt sind.

Q-Lernalgorithmus - I

Q und V^* stehen in enger Beziehung zueinander:

$$V^*(s) = \max_{a'} Q(s, a')$$

Damit läßt sich die Definition von $Q(s, a)$ umschreiben:

$$Q(s, a) \equiv r(s, a) + \gamma \max_{a'} Q(\delta(s, a), a')$$

Diese rekursive Definition von Q bildet die Basis für einen Algorithmus, der Q iterativ approximiert.

Q-Lernalgorithmus - II

Im folgenden sei $\hat{Q}$ der aktuelle Schätzwert für Q. s' sei der neue Zustand nach Ausführung der gewählten Handlung und r sei der dabei erzielte Reward.

Dann ergibt sich aus der rekursiven Definition von Q die Iterationsvorschrift wie folgt:

$$Q(s,a) \equiv r(s,a) + \gamma \max_{a'} Q(\delta(s,a), a')$$

$\Rightarrow$:

$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

Q-Lernalgorithmus - III

Damit lautet der Algorithmus:

1. Initialisierte alle Tabelleneinträge von $\hat{Q}$ zu 0.

2. Ermittle aktuellen Zustand s.

3. Loop

   - Wähle Handlung a und führe sie aus.
   - Erhalte Reward r.
   - Ermittle neuen Zustand s'.
   - $\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s',a')$
   - $s \leftarrow s'$.

   Endloop

Konvergenz des Q-Lernens

Satz: Es seien folgende Bedingungen erfüllt:

- $|r(s,a)| < \infty, \forall s, a$
- $0 \leq \gamma < 1$
- jedes (s,a)-Paar wird unendlich oft besucht

Dann konvergiert $\hat{Q}$ gegen die richtige Q-Funktion.

Kontinuierliche Systeme

Die Q-Funktion sehr großer oder kontinuierlicher Zustandsräume läßt sich nicht durch eine explizite Tabelle darstellen.

Man verwendet stattdessen einen Funktionsapproximations-Algorithmus, z.B. ein Neuronales Netz oder B-Splines.

Die Ausgaben des Q-Learning-Algorithmus dienen dem Neuronalen Netz als Trainingsbeispiele.

Konvergenz ist dann aber nicht mehr garantiert!

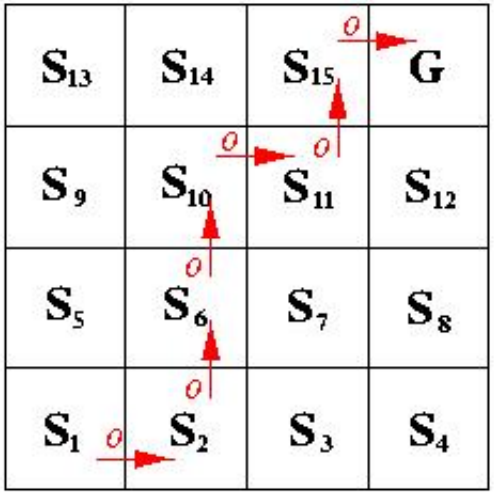
Beispiel: GridWorld - I

gegeben: $m \times n$-Grid

- $S = \{(x, y) | x \in \{1, \cdots, m\}, y \in \{1, \cdots, n\}\}$
- $A = \{up, down, left, right\}$
- $r(s, a) = \begin{cases} 100, \text{falls } \delta(s, a) = \text{Goalstate} \\ 0, sonst. \end{cases}$
- $\delta(s, a)$ gibt den Folgezustand entsprechend der durch a gegebenen Bewegungsrichtung an.

Beispiel: GridWorld - II

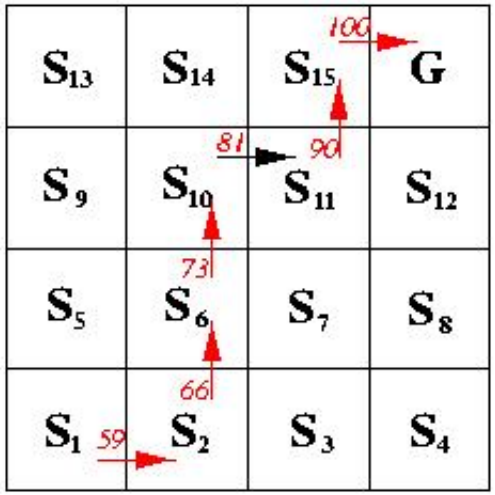
Beispiel für einen Pfad durch einen Zustandsraum:



Die Zahlen an den Pfeilen geben die momentanen Werte von $\hat{Q}$ an.

Beispiel: GridWorld - III

Entwicklung der $\hat{Q}$-Werte:



$$\hat{Q}(S_{11}, up) = \quad r + \gamma \max_{a'} \hat{Q}(s', a') = 0 + 0.9 * 100 = 90$$

$$\hat{Q}(S_{10}, right) = \quad 0 + 0.9 * 90 = 81$$

.
.
.

Q-Lernen - offene Fragen

- oft nicht vorhanden: Markov-Annahme, Beobachbarkeit
- kontinuierliche Zustand-Aktion-Räume
- Generalisierung über Zustand und Aktion
- Kompromiß zwischen “Exploration” und “Exploitation”
- Generalisierung der automatischen Bewertung (Kredit-Vergabe)

Visuell geführtes Greifen mittels selbstbewertendem Lernen

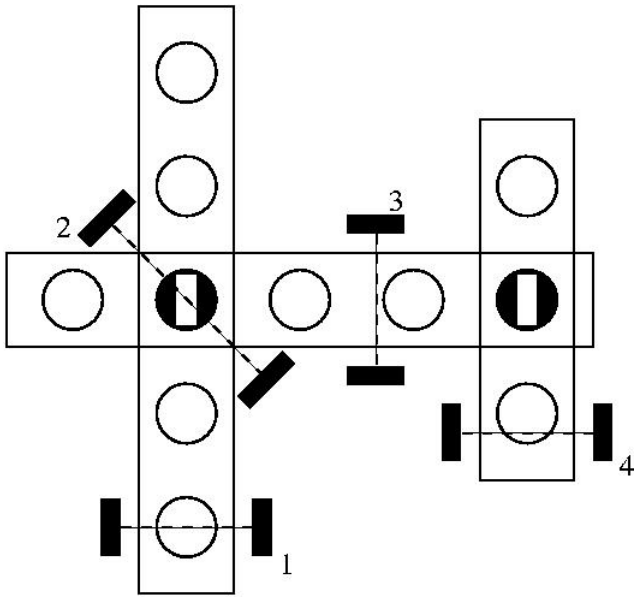
Griff ist optimal bzgl. lokaler Kriterien:

- Die Finger des Greifers können das Objekt am Greifpunkt umschließen
- Keine Reibung tritt zwischen Fingern und Objekt auf

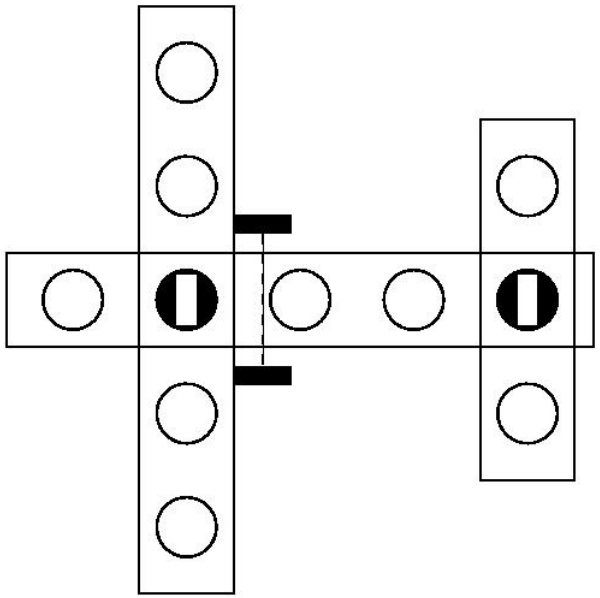
Griff ist optimal bzgl. globaler Kriterien:

- Kein bzw. minimales Drehmoment an den Fingern
- Objekt rutscht nicht aus dem Greifer
- Der Griff ist stabil, d.h. Objekt sitzt fest zwischen den Fingern

Bsp. Lokale und globale Kriterien



(a)



(b)

Zwei Ansätze

Ein Lerner:

- Die Zustände bestehen aus einem Satz $m + n$ lokaler und globaler Eigenschaften: $s = (f_{l_1}, \ldots, f_{l_m}, f_{g_1}, \ldots, f_{g_n})$.
- Der Lerner versucht sie auf die Aktionen $a = (x, y, \phi)$ abzubilden, wo x und y in Richtung x und y translationale Komponenten sind und ϕ die Rotation um den Annäherungsvektor des Greifers ist.

Zwei Lerner:

- Die Zustände für den ersten Lerner liefern nur die lokalen Eignschaften $s = (f_{l_1}, \ldots, f_{l_m})$.
- Der Lerner versucht sie auf Aktionen abzubilden, die nur aus der Rotationskomponente $a = (\phi)$ bestehen.
- Der zweite Lerner versucht, Zustände globaler Eigenschaften $s = (f_{g_1}, \ldots, f_{g_n})$ auf Aktionen translationaler Komponenten abzubilden: $a = (x, y)$.

Aufbau

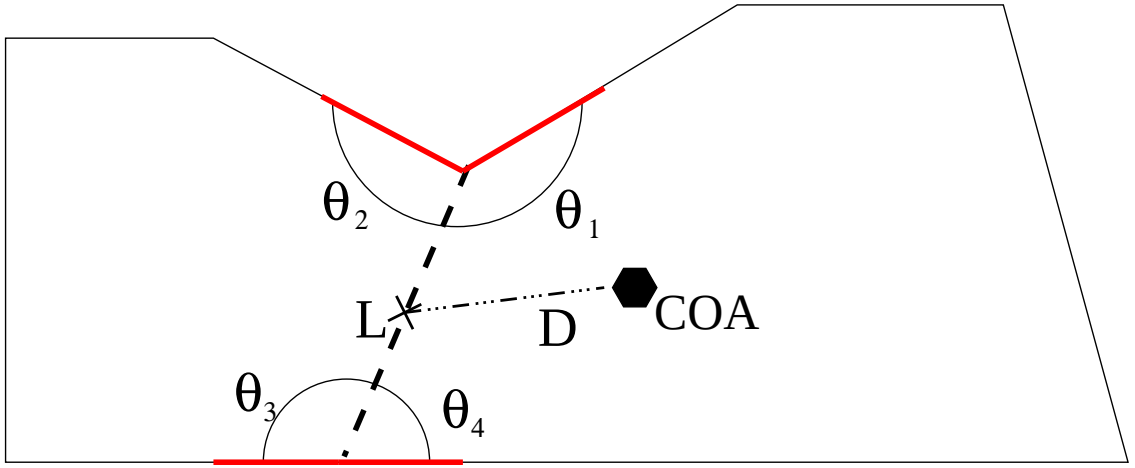
- Zweikomponentiges Lernsystem:

1. Orientierungs-Lerner	**2. Orts-Lerner**
Operiert auf lokalen Kriterien	Operiert auf globalen Kriterien
Gleich für jedes Objekt	Unterschiedlich für jedes neue Objekt

- Einsatz von Multisensorik:
  - Kamera
  - Kraft-Moment-Sensor

- Beide Lerner arbeiten aufeinanderfolgend im Perzeptions-Aktions-Zyklus

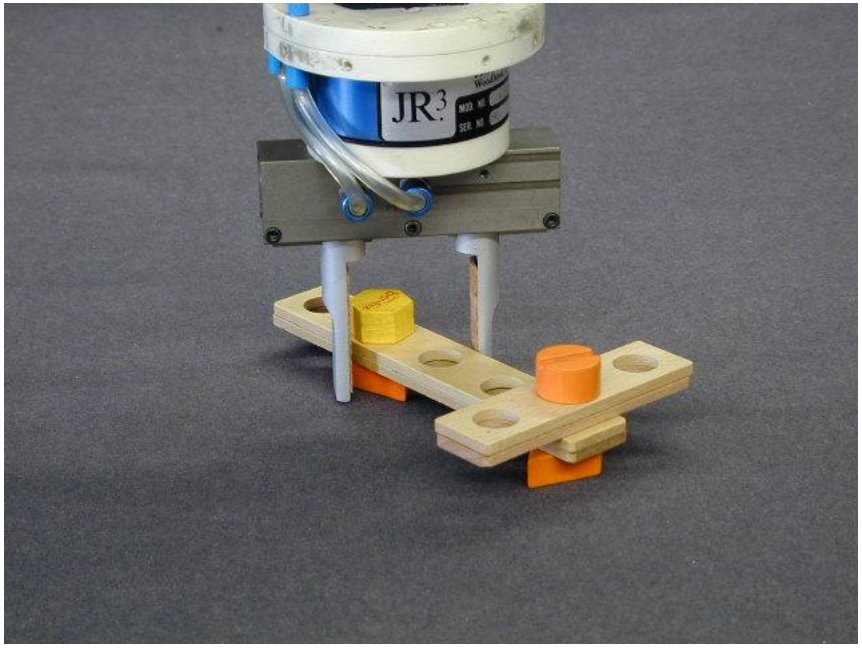
Zustandskodierung

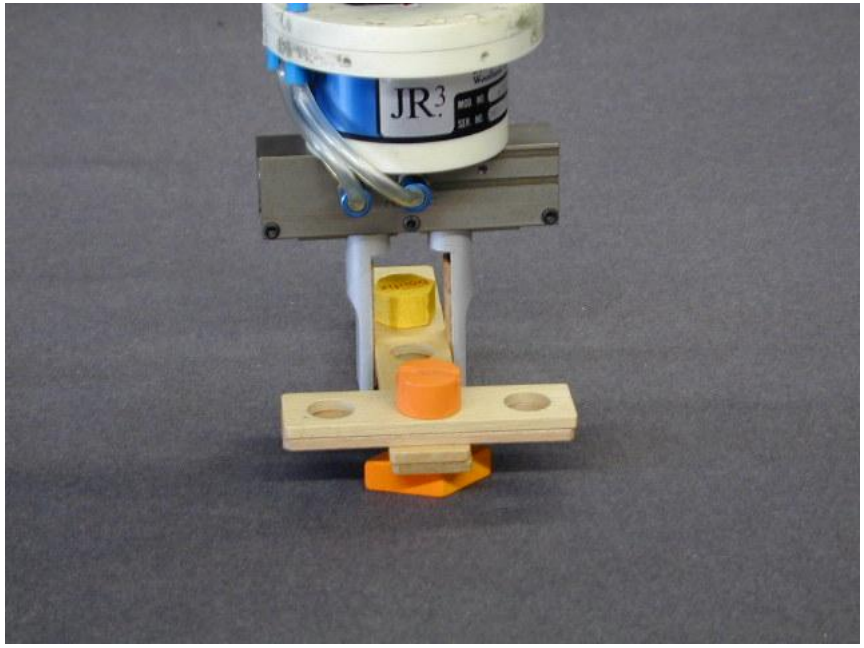


Der Orientierungs-Lerner benutzt Länge L sowie Winkel $\Theta_1, \ldots, \Theta_4$ wärend der Orts-Lerner die Distanz D zwischen Mittelpunkt der Greiflinie und Bildschwerpunkt des Objektes nutzt.

Selbstbewertungsmaßnahmen im Orientierungs-Lerner

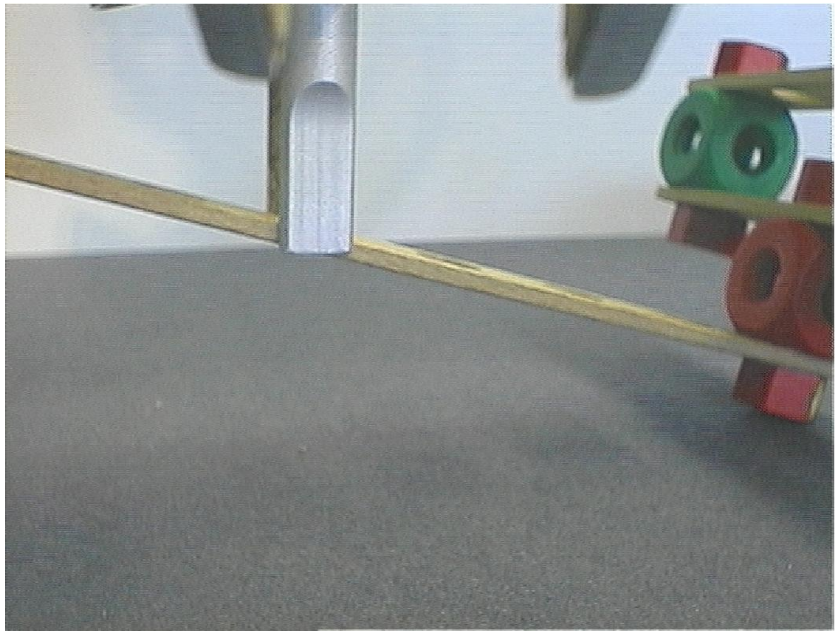
Visuelle Bewertung des Grifferfolges:
Reibung resultiert in Rotation oder Versatz des Objektes

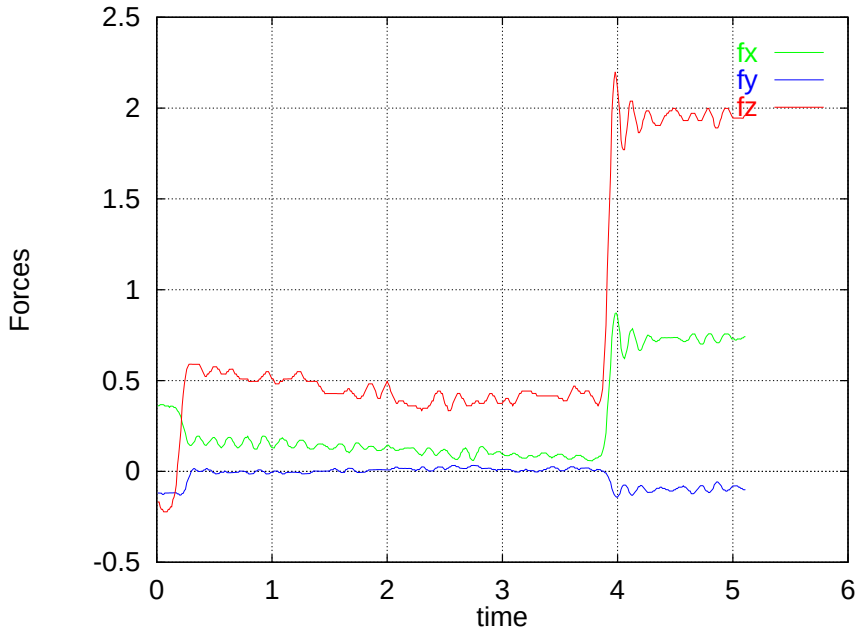




Selbstbewertungsmaßnahmen im Orts-Lerner I

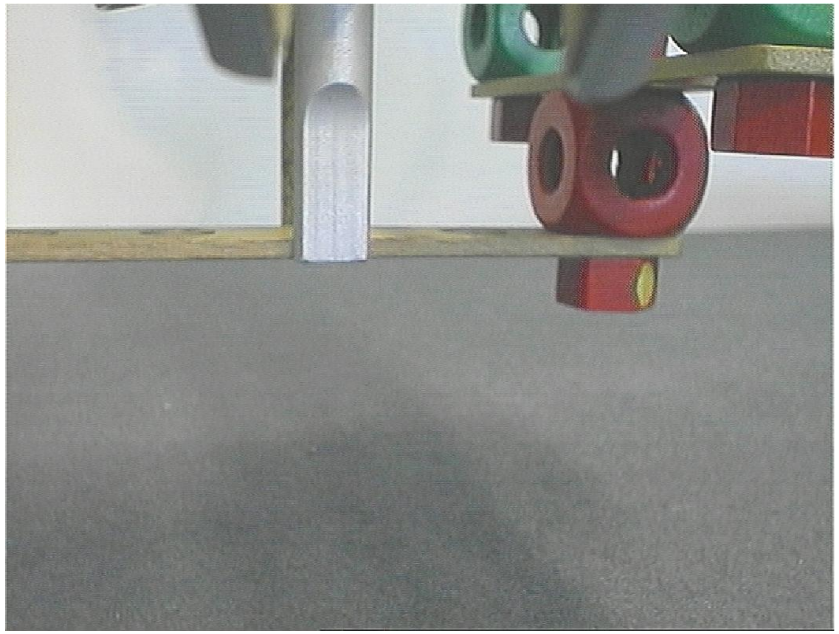
Bewertung mittels Kraftmomenten-Sensors:
Instabiler Griff – analysiert durch Kraftmessung

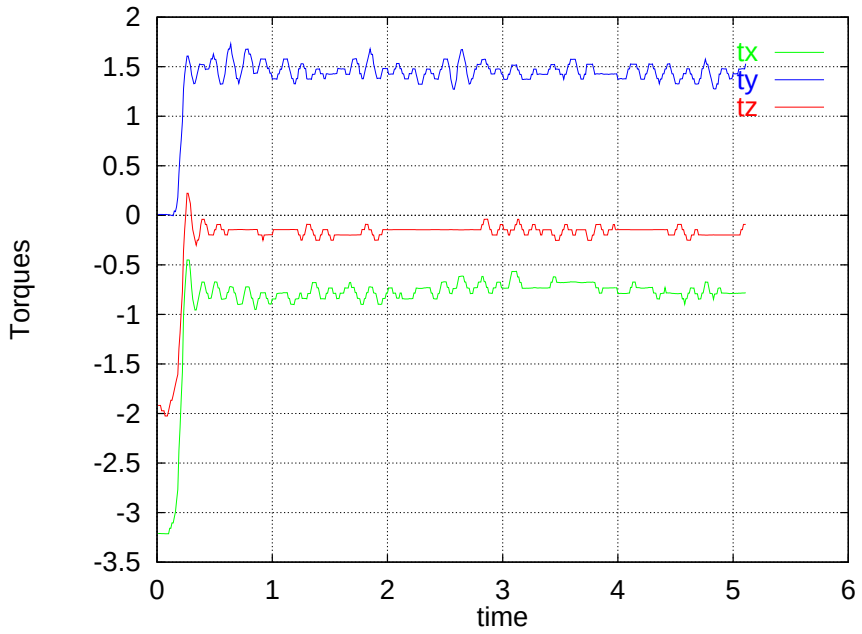




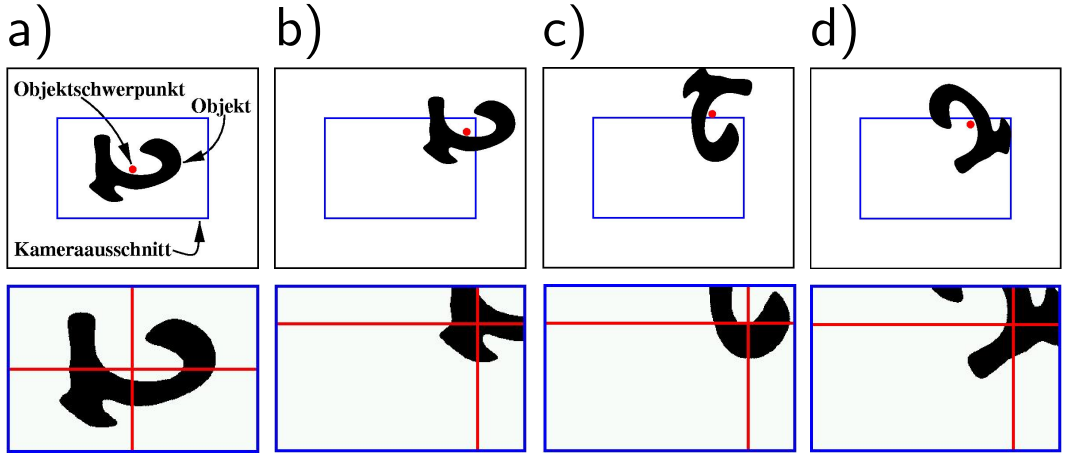
Selbstbewertungsmaßnahmen im Orts-Lerner II

Suboptimaler Griff – analysiert über Drehmomente





Das Problem der unvollständigen Zustandsinformation

Man spricht auch von **verborgenen Zuständen** (engl.: *hidden states*).

Beispiel für unvollständige Zustandsinformation: