
2. Reaktive Agenten

Prof. Dr. Rudolf Kruse

University of Magdeburg

Faculty of Computer Science

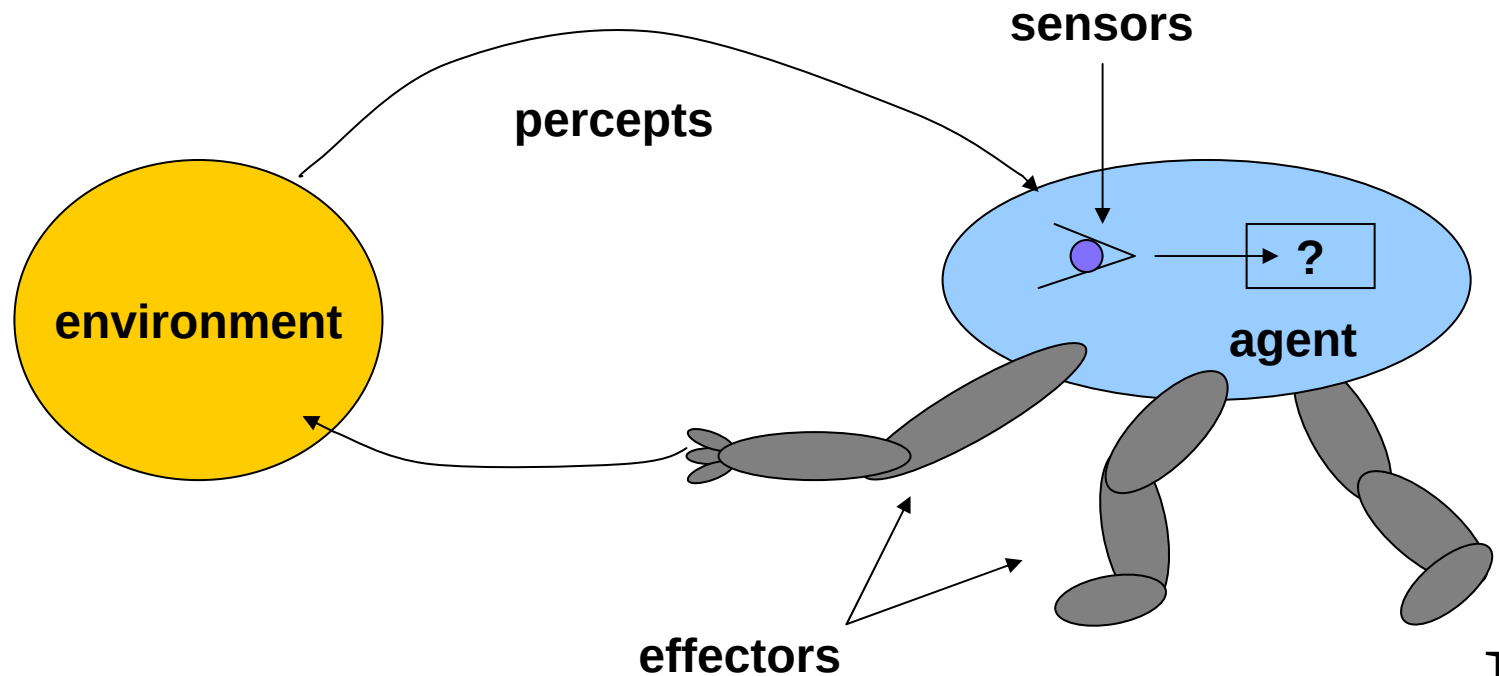
Magdeburg, Germany

rudolf.kruse@cs.uni-magdeburg.de



Agenten

Ein intelligenter Agent interagiert mit seiner Umgebung mittels Sensoren und Effektoren und verfolgt gewisse Ziele:



Agenten

Beispiele:

- **Menschen und Tiere**
- **Roboter und Software-Agenten (Softbots)**
- **aber auch: Heizungen, ABS...**

Agenten

- Es gibt jedoch andere Definitionen aus den unterschiedliche Fachgebieten wie
 - Ein Programm ist ein Softwareagent, wenn es korrekt in einer (Agenten-) Sprache wie ACL, KQML oder KIF kommuniziert. BDI Agenten werden durch Überzeugungen (*believes*), Wünsche (*desires*) und Intentionen (*intentions*) beschrieben, praktisch werden sie mit einer Modallogik und speziellen Datenstrukturen implementiert.

Agenten

- Beispiel 1: Simulation Soccer
- RoboCup – Roboterfußball



Agenten

Beispiel 2: Taxifahrer

Typ	Taxifahrer
Wahrnehmung	Kameras, Tachometer, GPS, Mikrophon
Aktionen	Steuern, Schalten, Bremsen, mit Gästen sprechen
Ziele	Sichere, schnelle, legale, komfortable Fahrt; Profit maximieren
Umgebung	Straßen, andere Verkehrsteilnehmer, Fußgänger, Fahrgäste

Charakterisierung von Agenten

- **Agenten können charakterisiert werden durch:**
 - Wahrnehmungen (**perceptions**)
 - Aktionen (**actions**)
 - Ziele (**goals**)
 - Umgebung (**environment**)
 - (= PAGE)

Beispiele von Agenten nach PAGE

Art	Wahrnehmung	Aktionen	Ziele	Umgebung
Medi-zinisches Diagnose-system	Symptome, Diagnose, Antworten des Patienten	Fragen, Tests, Behand-lungen	Gesundheit,geringe Kosten	Patient, Hospital
Satelliten-bildanalyse	Punkte verschiedener Intensität	Klassifikation	Korrekte Klassifikation	Satelliten-bilder
Roboter	Punkte verschiedener Intensität	Teile aufheben und einsortieren	Teile richtig einsortieren	Förderband mit Teilen

Beispiele von Agenten nach PAGE

Art	Wahrnehmung	Aktionen	Ziele	Umgebung
Raffinerie Regler	Temperatur, Druck	Öffnen, Schließen von Ventilen, Temperatur einstellen	Reinheit, Ertrag, Sicherheit maximieren	Raffinerie
Interaktiver Englisch-tutor	Eingegebene Wörter und Übungen, Vorschläge	Korrekturen ausgeben	Testergebnisse des Studenten maximieren	Menge von Studenten

Typen von Agenten (I)

Man kann Agenten nach der Art und Weise ihrer Umwelt-Interaktionen unterscheiden:

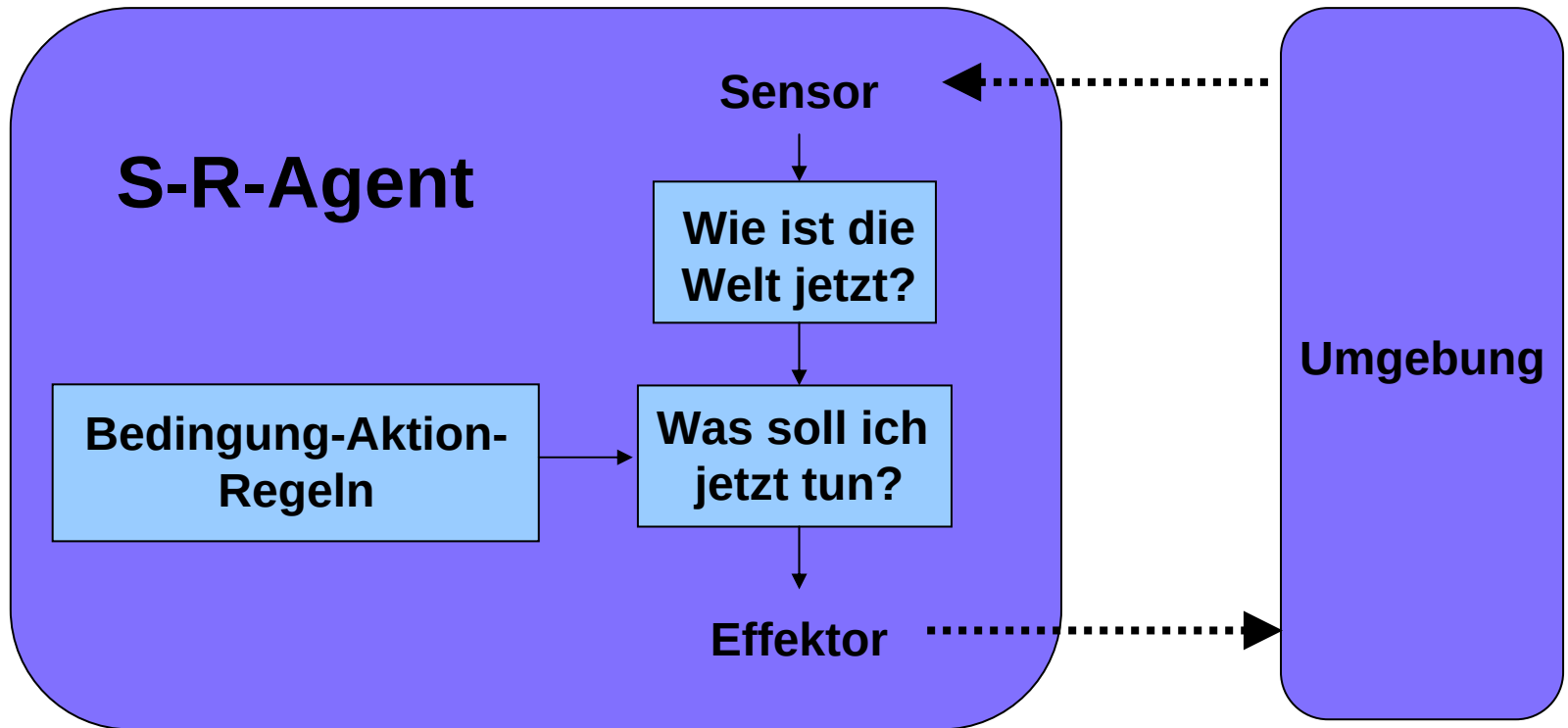
- **reaktive Agenten**
 - **die über ein Reiz-Antwort-Schema ihr Verhalten steuern,**
- **reflektive Agenten**
 - **die planbasiert agieren, also explizit Pläne, Ziele und Intentionen verarbeiten,**
- **situierte Agenten**
 - **die einfaches Reagieren und überlegtes Handeln in dynamischer Umwelt verbinden,**

Typen von Agenten (II)

- autonome Agenten
 - **die zwischen reflektiven und situierten Agenten einzuordnen sind (und meist in der Robotik verwendet werden),**
- rationale Agenten
 - **die den reflektiven Agenten entsprechen, jedoch eine ausgeprägte Bewertungsfunktionalität besitzen,**
- soziale Agenten
 - **die in der Lage sind, ihr Handeln an einem Gemeinziel auszurichten.**

Stimulus-Response Agent

- Der einfache reaktive Agent antwortet unmittelbar auf Wahrnehmungen (Stimulus-Response Agent)



Stimulus-Response Agent

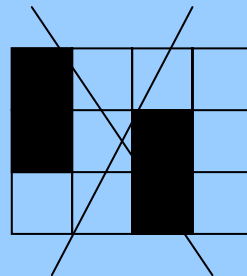
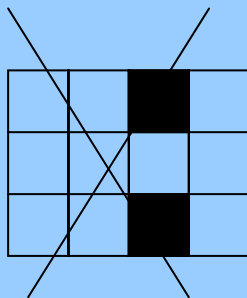
Das Skelett eines solchen Agenten sieht etwa so aus:

```
function S-R-Agent (percept) returns action
  static : rules
  state INTERPRET-INPUT (percept)
  rule RULE-MATCH (state, rule)
  action      RULE-ACTION (rule)
  return      ← action
```

Der Agent sucht eine Regel deren Bedingung der gegebenen Situation entspricht und führt die zugehörige Aktion (Regel-Konklusion) aus.

Beispiel: Gitterwelt (*Grid-Space World*)

Die Umwelt sei der Einfachheit halber eine fiktive zweidimensionale Gitterzelleneinheit, in der sich verschiedene (Spielzeug-)Agenten tummeln. In den Zellen können Objekte mit verschiedenen Eigenschaften sein, es gibt auch Barrieren. Agenten können von Zelle zu Zelle laufen. Es gibt keine „tight spaces“ (Lücken zwischen Objekten und Begrenzungen, die nur eine Zelle breit sind.)



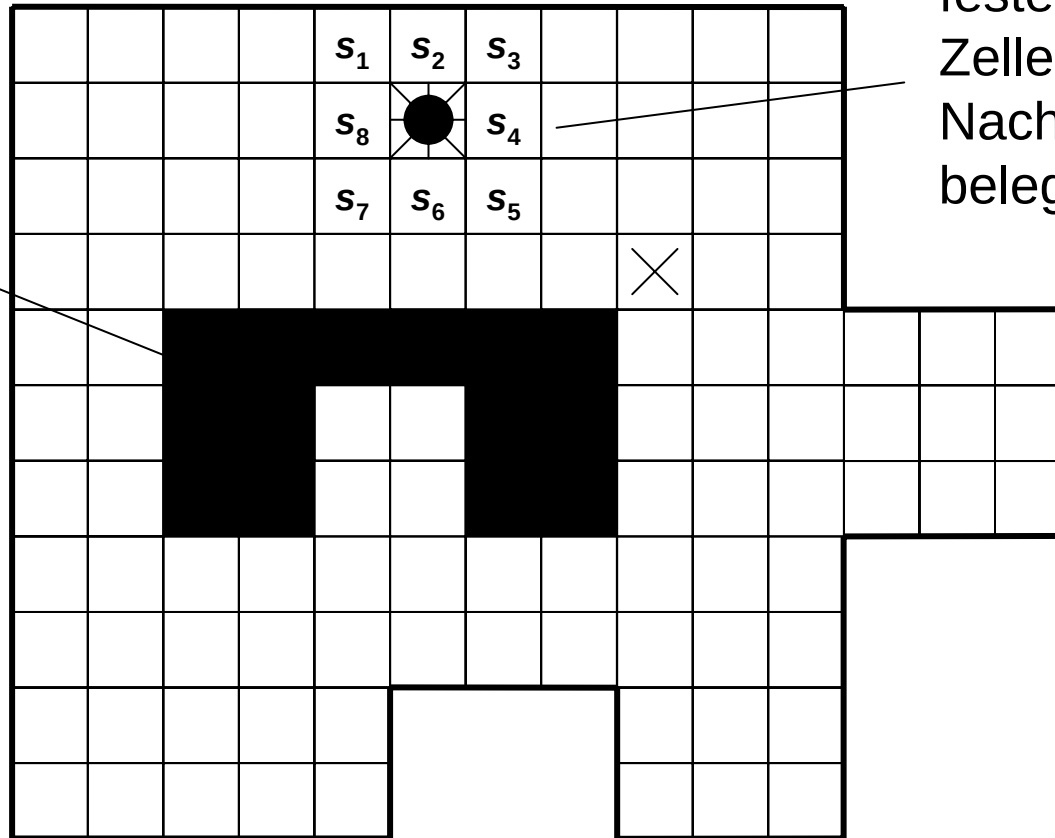
Diese Umgebungen
sind nicht erlaubt

Gitterwelt (II)

Begrenzung

Der Roboter kann feststellen, welche Zellen in seiner Nachbarschaft belegt sind.

Massives Objekt

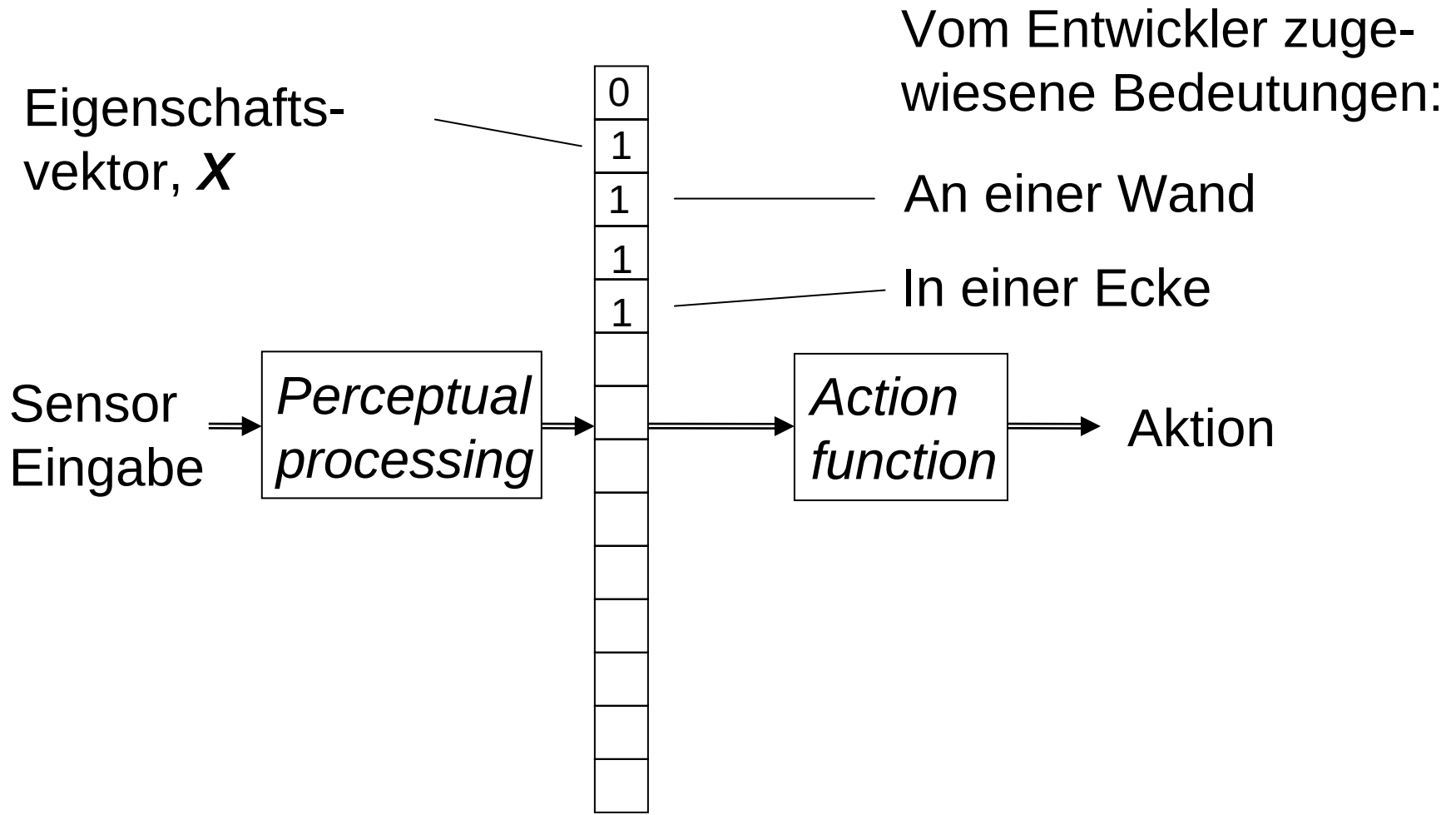


Beispiel: Roboter in zweidimensionaler Gitterwelt

Gitterwelt (III)

- $s_i \in \{0,1\}$ Sensoreingabe:
 - $s_j = 0 \Leftrightarrow$ Zelle s_j ist frei für Roboter
 - Bei X im Bild ist Sensoreingabe (0,0,0,0,0,0,1,0)
- Es gibt die 4 Aktionen:
 - north, east, south, west
 - north bewegt z.B. den Roboter eine Zelle nach oben, falls die Zelle frei ist, ansonsten wird der Roboter nicht bewegt
- Eine Aufgabe wird häufig in 2 Schritten, der *perception processing* und der *action computation* Phase gelöst

Perception und Action Komponenten



Beispiel: Wandverfolgung

- Aufgabe: Gehe zu einer Zelle an der Begrenzung eines Objekts und folge dieser Grenze
- *Perception:*
 - Es gibt 2^8 verschiedene Sensoreingaben (einige fallen wegen „no tight Space“ Voraussetzung weg).
 - Wir wählen 4 Merkmale $x_1, \dots, x_4$:

$$x_1 = 1 \Leftrightarrow (s_2 = 1 \vee s_3 = 1)$$

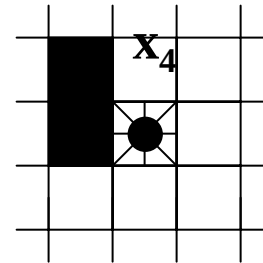
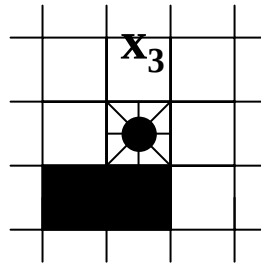
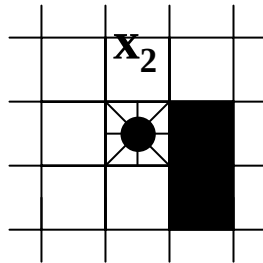
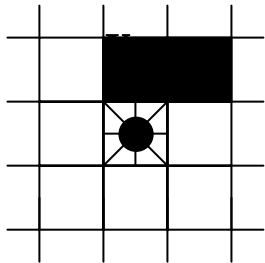
$$x_2 = 1 \Leftrightarrow (s_4 = 1 \vee s_5 = 1)$$

$$x_3 = 1 \Leftrightarrow (s_6 = 1 \vee s_7 = 1)$$

$$x_4 = 1 \Leftrightarrow (s_8 = 1 \vee s_1 = 1)$$

Wandverfolgung (II)

- Das Merkmal in jedem Diagramm hat genau dann den Wert 1, wenn mindestens eine der markierten Zellen belegt ist.



(In komplexen Welten sind Informationen typischerweise unsicher, vage, oder sogar falsch.)

Wandverfolgung (III)

■ Aktionen:

- Falls keines der 4 Merkmale den Wert 1 hat, führe north durch
- Sonst:

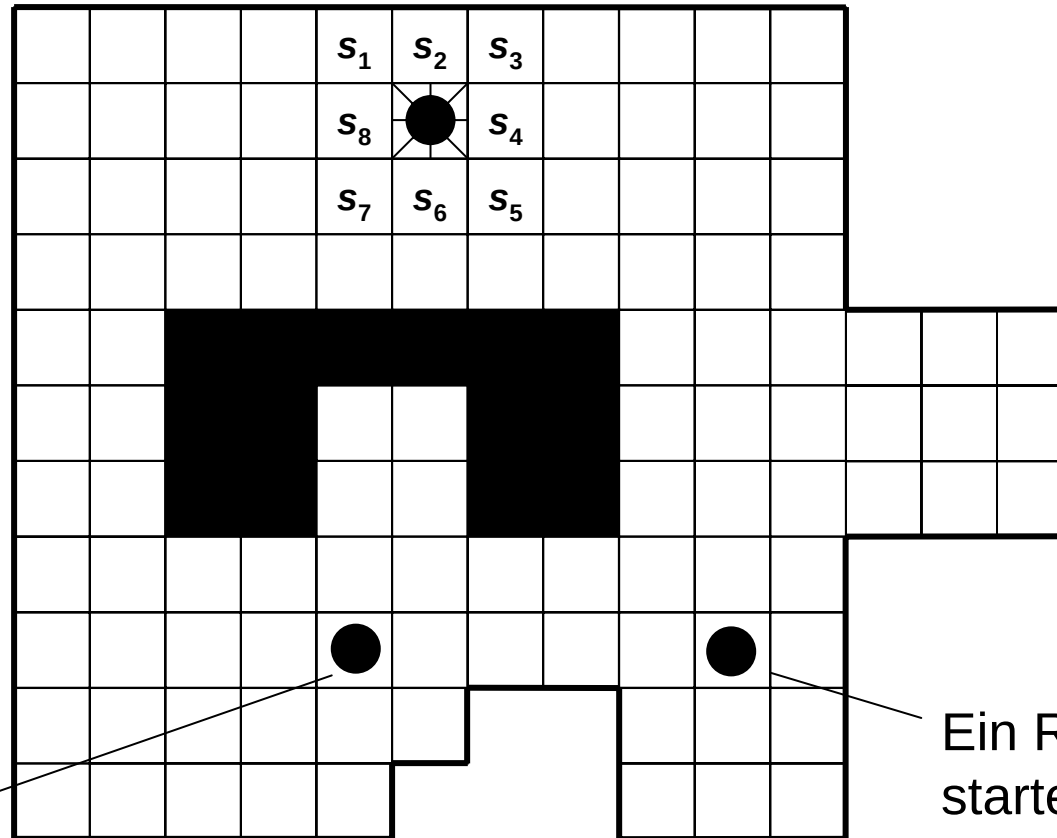
If $x_1=1$ and $x_2=0$ then east

If $x_2=1$ and $x_3=0$ then south

If $x_3=1$ and $x_4=0$ then west

If $x_4=1$ and $x_1=0$ then north

Wandverfolgung (IV)



Ein Roboter der hier startet bewegt sich entgegen dem Uhrzeigersinn am Objekt entlang.

Ein Roboter der hier startet bewegt sich im Uhrzeigersinn an der äusseren Begrenzung entlang.

Auswertung der Sensoreingaben

- Für die beiden Phasen *perception processing* und *action computation* werden oft Boolesche Algebren verwendet.
- So gilt:

$$x_4 = s_1 \vee s_8 \text{ und go north}$$

$$\Leftrightarrow$$

$$(\bar{x}_1 \wedge \bar{x}_2 \wedge \bar{x}_3 \wedge \bar{x}_4) \vee (x_4 \wedge \bar{x}_1) = 1$$

Auswertung der Sensoreingaben

- Eine geeignete Repräsentationsform für Aktionen sind Regelsysteme der Form $c_j \rightarrow a_j$, wobei
 - c_j der Bedingungsteil und
 - a_j der Aktionsteil sind.
- In unserem Beispiel erhält man z. B. die Regeln:

$$x_4 \wedge \bar{x}_1 \rightarrow go\ north$$

$$1 \rightarrow go\ north$$

$$x_3 \wedge \bar{x}_4 \rightarrow go\ west$$

$$x_2 \wedge \bar{x}_3 \rightarrow go\ south$$

$$x_1 \wedge x_2 \rightarrow go\ east$$

- Regelsysteme und Boolesche Algebren kann man gut mit Hilfe von Netzwerken implementieren.

Wandverfolgung

- Ein alternativer Ansatz zu S-R-Agenten besteht in der Einführung von Subsumptions–Modulen:
 - Jedes Modul enthält Sensorinformationen direkt von der Umwelt.
 - Sind die spezifizierten Voraussetzungen des Moduls erfüllt, so wird das Programm ausgeführt.
 - „Höhere“ Module subsumieren „tiefere“, d.h. falls die Voraussetzung eines höheren Moduls erfüllt ist, wird das tiefere Modul durch das höhere ersetzt.

Wandverfolgung

Sensorsignale

