Intelligente Systeme

Einführung

Prof. Dr. Rudolf Kruse Georg Ruß
Christian Moewes

{kruse,russ,cmoewes}@iws.cs.uni-magdeburg.de

Arbeitsgruppe Computational Intelligence
Institut für Wissens- und Sprachverarbeitung
Fakultät für Informatik
Otto-von-Guericke Universität Magdeburg

Details zur Lehrveranstaltung

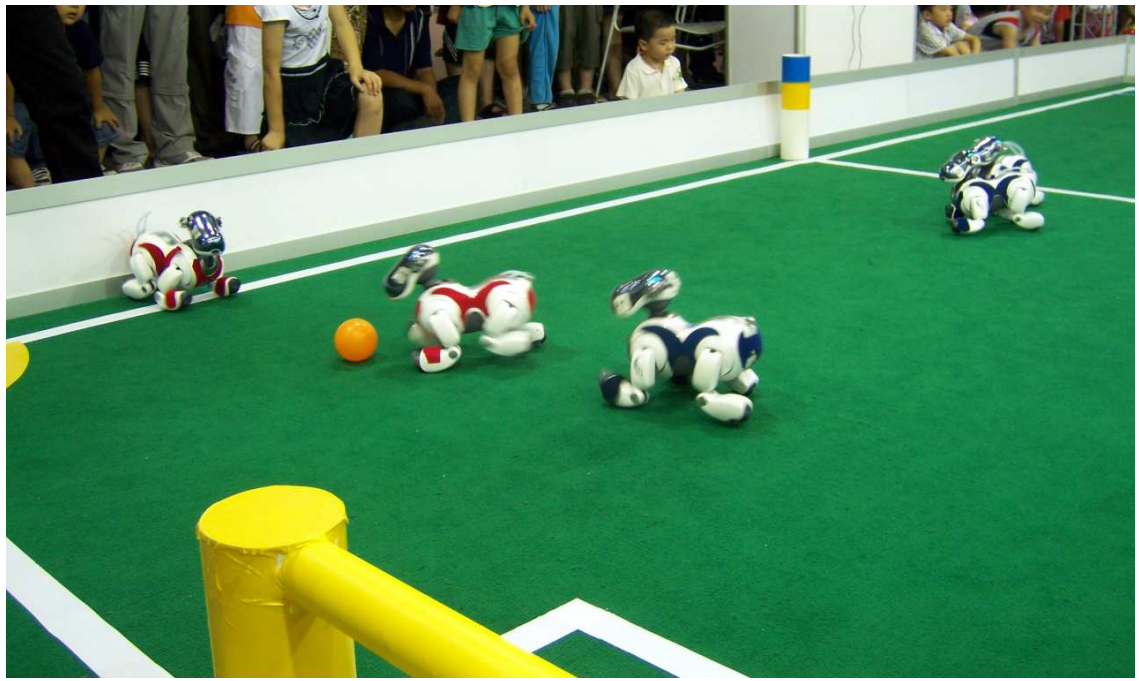
- Vorlesung
  - Konsultation: Mittwoch, 11:00 – 12:00, G29-008
  - Bevorzugt per e-mail: `kruse@iws.cs.uni-magdeburg.de`
- Übungen
  - Übungsleiter: Georg Ruß
  - G29-013, `russ@iws.cs.uni-magdeburg.de`
- Aktuelle Informationen
  - `http://fuzzy.cs.uni-magdeburg.de/`

Intelligente Systeme beim “Robocup”

Robocup

Ziel:

> *“By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team.”*



Intelligente Systeme beim “Robocup”

Humanoider Roboter

weitere Informationen zum Robocup: `http://www.robocup.org`

Intelligente Systeme bei der “DARPA Grand Challenge”

DARPA Grand Challenge

Ziel:

> *The DARPA Urban Challenge is an autonomous vehicle research and development program with the goal of developing technology that will keep warfighters off the battlefield and out of harm's way. The Urban Challenge features autonomous ground vehicles maneuvering in a mock city environment, executing simulated military supply missions while merging into moving traffic, navigating traffic circles, negotiating busy intersections, and avoiding obstacles.*

(von `http://www.darpa.mil/grandchallenge/overview.asp`)

Intelligente Systeme bei der “DARPA Grand Challenge”

Zwei (von elf) Teams: *Stanford Racing* und *Victor Tango*

Bild von `http://www.darpa.mil/GRANDCHALLENGE/gallery.asp`

“Geschäftsmodell” der Arbeitsgruppe CI

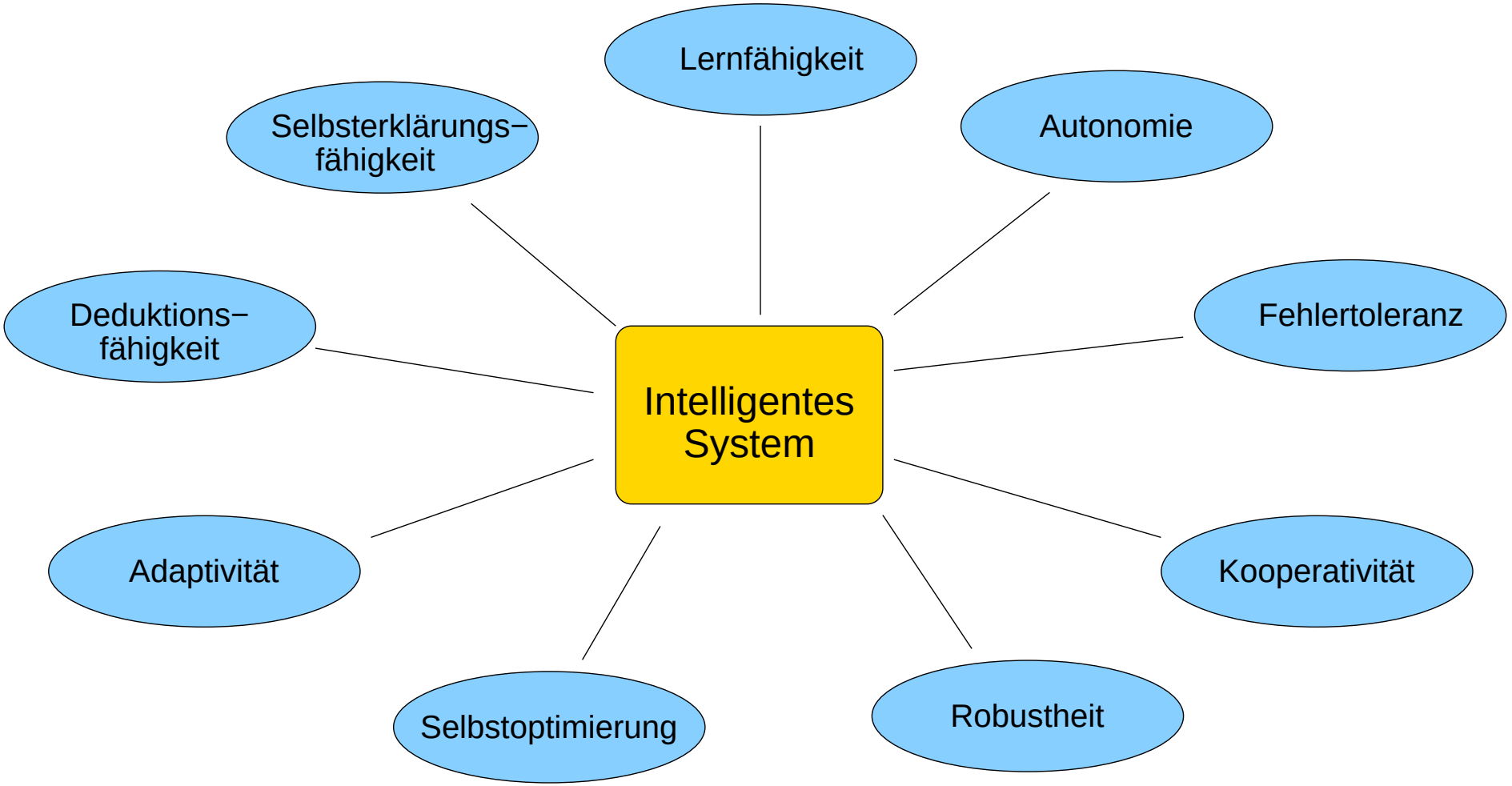
- Entwicklung neuer Methoden in Forschungsprojekten
- Entwicklung frei verfügbarer Analyse-Software
- Nutzung in Industrieprojekten, derzeit:
  - Teilebedarfsplanung mit Markov-Netzen (VW)
  - Information Mining (BMW, Daimler)
  - Bayes-Methoden im Finanzwesen (Deutscher Giro- und Sparkassenverband)
  - Kundenzufriedenheitsanalyse (BT)

Was ist ein *Intelligentes System*?

Intelligente Systeme

- Was ist ein intelligentes System?
  - Wissensverarbeitung wird benötigt, um künstliche intelligente Systeme zu entwickeln
- Begriffsbestimmungen:
  - Daten: Zeichen, die maschinell verarbeitet werden können
  - Information: durch entsprechende Interpretation erhalten Daten einen Sinn; Information ist der abstrakte Inhalt von Daten
  - Wissen: verknüpft Informationen sinngebend miteinander
- Merkmale intelligenter Systeme:
  - Lernfähigkeit
  - Autonomie
  - Fehlertoleranz
  - ...

Merkmale Intelligenter Systeme



Was ist ein Intelligentes System?

- Der Begriff *Intelligentes System* wandelt sich mit dem wissenschaftlichen Fortschritt
  - 1967: Taschenrechner
  - Programme zur symbolischen Integration
  - 1997: Schachcomputer DEEP BLUE: gewann gegen Kasparov 3.5 zu 2.5
  - 1998: Automatikgetriebe AG4 für VW New Beetle
  - 2002: Schachcomputer DEEP FRITZ: gegen Vladimir Kramnik 4 zu 4
  - Expertensysteme (wissensbasierte Systeme, intelligente Systeme)
  - VW Touareg Stanley bei DARPA Grand Challenge 2006
  - ...

Was ist ein Intelligentes System?

- Ein Problem sind oft anmaßende Voraussagen über Fortschritte der Künstlichen Intelligenz (KI)
- H. A. Simon (Nobelpreisträger 1978) und A. Newell (Turing-Award 1975) behaupten 1957, dass im Jahre 1967
  - ein Rechner Schachweltmeister sein wird,
  - ein Computer einen wichtigen neuen mathematischen Satz entdecken und beweisen wird,
  - ein digitaler Rechner ein Musikstück schreiben wird, welchem von Kritikern ein beachtlicher ästhetischer Wert bescheinigt wird.

Was ist ein Intelligentes System?

Realistischere Einschätzungen

- Im deutschen Delphi-Report 1993 steht, dass im Jahre 2005
  - Rechner entwickelt werden, die ungenaue Informationen in einer Art gesunden Menschenverstandes verarbeiten können,
  - Informationsdatenbanken eingesetzt werden, die durch automatisches Lernen ihr Wissen vermehren,
  - tragbare automatische Übersetzungsgeräte (einfache, alltägliche Konversation in beiden Richtungen) mit Spracheingaben kommerzialisiert werden,
  - in Büros Geräte weitverbreitet sind, die Texte in handschriftlicher Fließschrift lesen können

Was ist ein Intelligentes System?

- Es gibt viele Herangehensweisen, das Gebiet ist hochgradig interdisziplinär
  - *kognitiver Ansatz:* Simulation kognitiver Prozesse, Analyse menschlicher Denkweise, *general problem solver*
  - *ingenieurwissenschaftlicher Ansatz:* Konstruktion von Systemen, die gewisse menschliche Wahrnehmungs- und Verstandsleistungen maschinell verfügbar machen (Produkte wie Gesichtserkenner, Roboter, Kooperation mit Gehirnforschern, . . .)

Was ist ein Intelligentes System?

- Es gibt spannende philosophische Fragestellungen, wie beispielsweise *Can machines think?*

“can”
Es gibt mehrere Bedeutungen: (“Kann denken”) heute – irgendwann – im Prinzip; tatsächlich ist derzeit die Frage “unentscheidbar”, ob man Systeme mit menschenähnlichen Fähigkeiten wird bauen können; wir sind mit den Fortschritten auf dem Weg dahin zufrieden und verdienen Geld sowie Ruhm mit innovativen Produkten.

Was ist ein Intelligentes System?

"machine"

- Es muss kein Stahlroboter sein, es kann auch ein biologischer Mechanismus sein (Bakterium Haemophilus influenzae Rd hat 10^7 Basenpaare, 1743 Gene, ...).
- Was passiert, wenn das menschliche Genom entziffert und verstanden ist?
- Des Weiteren: Bewusstseinsdiskussion, Leib-Seele-Problem, das "ich" im Gehirn

Was ist ein Intelligentes System?

"think"

- Menschen sind Maschinen, also können Maschinen denken
- Searle (1992): Denken funktioniert nur in speziellen (tatsächlich lebenden) Maschinen
- Newell and Simon (1976): Physical symbol system hypothesis
  - ein PSS hat die notwendigen und hinreichenden Bedingungen für intelligentes Verhalten
  - ein PSS ist eine Maschine, die symbolische Daten manipulieren kann (z.B. Computer)
- Kohonen et al. (1980): Zur Entwicklung intelligenter Maschinen benötigt man subsymbolische Prozesse (z.B. Signale)
- Zadeh (1964): wirklich intelligente Systeme müssen eine Art *fuzzy logic* benutzen (keine binäre Logik)
- Turing-Test (1950)

Turing-Test (1950)

It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. He communicates with them via teletype. The objective of the game for the interrogator is to determine which of the other two is the man and which is the woman. He knows them by labels X and Y, and at the end of the game he says either "X is A and Y is B" or "X is B and Y is A". The interrogator is allowed to put questions to A and B thus:

> *C: Will X please tell me the length of his or her hair?*

Now suppose X is actually A, then A must answer. It is A's objective in the game to try and cause C to make the wrong identification. . . .

Turing-Test (1950)

The objective of the game for the third player (B) is to help the interrogator.

...

We now ask the question, “What will happen when a machine takes the part of A in this game?” Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, “Can a machine think?”

The Turing test is often simplified to one in which a machine attempts to convince a human interrogator that it is a human.

Was ist ein Intelligentes System?

- Das Gebiet Wissensverarbeitung ist extrem innovativ:
  - objektorientierte Programmiersprachen
  - graphische Oberflächen
  - Expertensysteme
  - Software-Agenten (Internet)
  - Autonome Roboter

  sind hier erfunden worden.

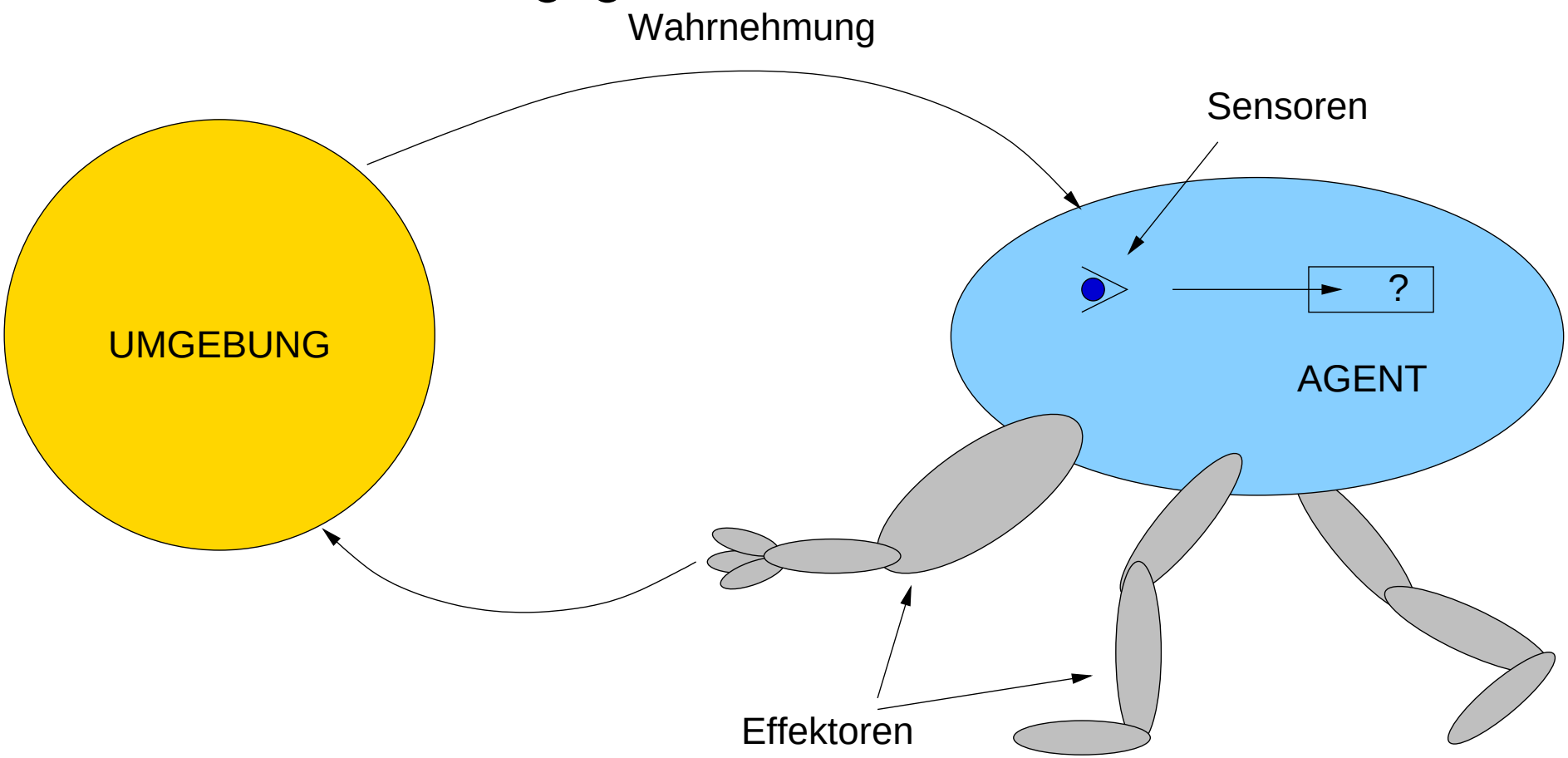
Reaktive Agenten

Gliederung der Vorlesung

1. **Reaktive Agenten**
2. **PAGE-Prinzip**
3. **Stimulus-Response-Agenten**

Agenten

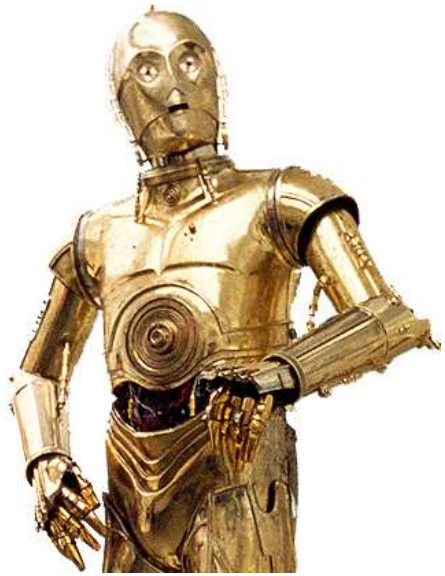
Ein intelligenter Agent interagiert mit seiner Umgebung mittels Sensoren und Effektoren und verfolgt gewisse Ziele:



Agenten

Beispiele

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

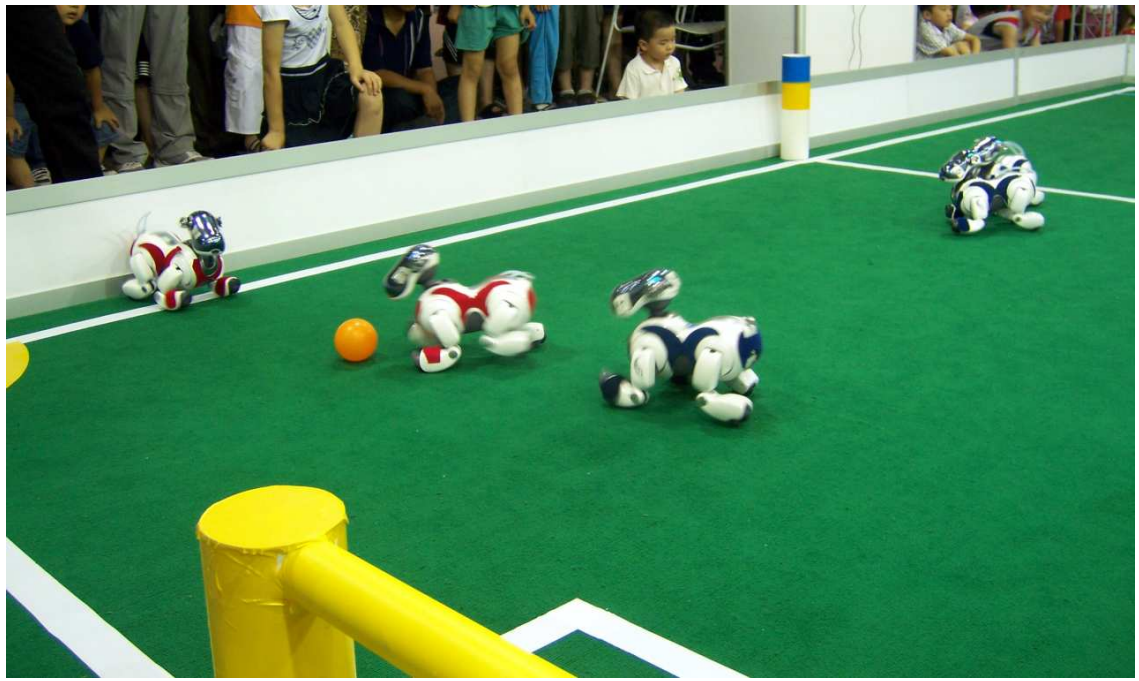


Agenten

- Es gibt jedoch andere Definitionen aus den unterschiedlichen Fachgebieten wie beispielsweise:
  - Ein Programm ist ein Softwareagent, wenn es korrekt in einer (Agenten-)Sprache wie ACL, KQML oder KIF kommuniziert.
    BDI-Agenten werden durch Über-zeugungen (*beliefs*), Wünsche (*desires*) und Absichten (*intentions*) beschrieben; praktisch werden sie mit einer Modallogik und speziellen Datenstrukturen implementiert.

Agenten

- Beispiel: Simulation Soccer
- RoboCup: Roboterfußball



Beispiel 2: Taxifahrer

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

Gliederung der Vorlesung

1. Reaktive Agenten
2. **PAGE-Prinzip**
3. Stimulus-Response-Agenten

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**
Medizinisches Diagnosesystem	Symptome, Diagnose, Antworten des Patienten	Fragen, Tests, Behandlungen	Gesundheit, geringe Kosten	Patient, Krankenhaus
Satellitenbildanalyse	Punkte verschiedener Intensität	Klassifikation	Korrekte Klassifikation	Satellitenbilder
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 Englischtutor	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:
  - steuern über ein Reiz-Antwort-Schema ihr Verhalten
- reflektive Agenten:
  - agieren planbasiert, verarbeiten also explizit Pläne, Ziele und Intentionen
- situierte Agenten:
  - verbinden einfaches Reagieren und überlegtes Handeln in einer dynamischen Umwelt

Typen von Agenten (II)

- autonome Agenten:
  - sind zwischen reflektiven und situierten Agenten einzuordnen (werden meist in der Robotik verwendet)
- rationale Agenten:
  - entsprechen den reflektiven Agenten, allerdings mit ausgeprägter Bewertungsfunktionalität
- soziale Agenten:
  - sind in der Lage, ihr Handeln an einem Gemeinziel auszurichten

Gliederung der Vorlesung

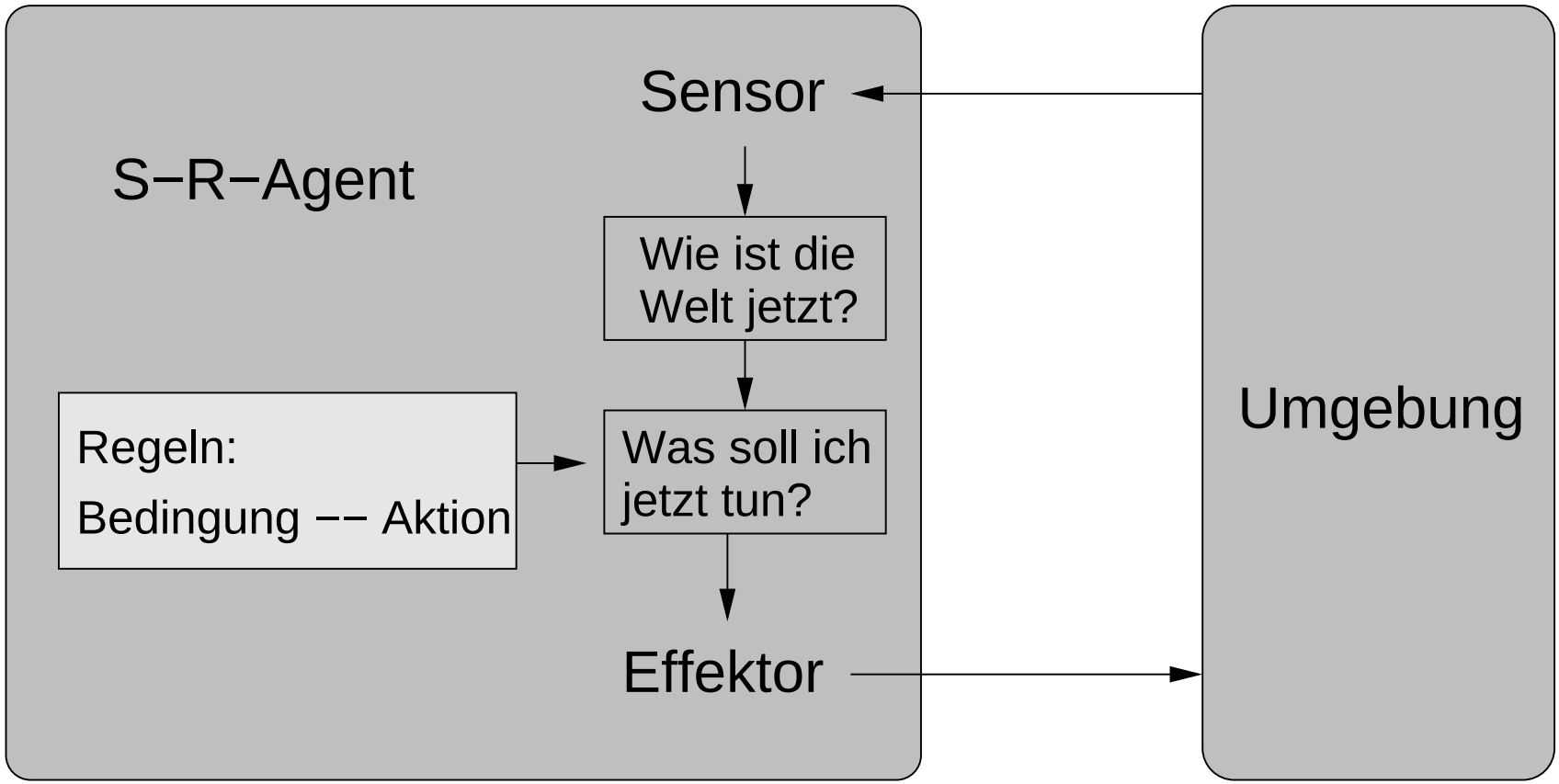
1. Reaktive Agenten

2. PAGE-Prinzip

3. **Stimulus-Response-Agenten**
   - Gitterwelt
   - Signalverarbeitung
   - Beispiel: Wandverfolgung

Stimulus-Response-Agent

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



Stimulus-Response-Agent

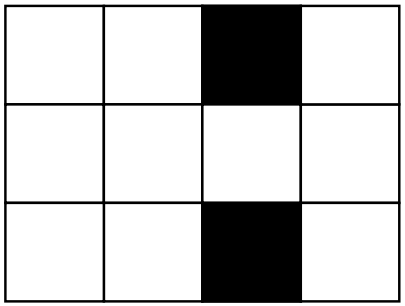
Das Skelett eines solchen Agenten sieht etwa so aus:

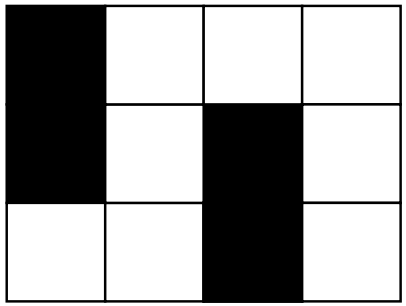
```
function S-R-Agent(percept) returns action
     static: rules
     state INTERPRET-INPUT(percept)
     rule RULE-MATCH(state, rule)
     action RULE-ACTION(rule)
     return action
```

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

Gitterwelt (I)

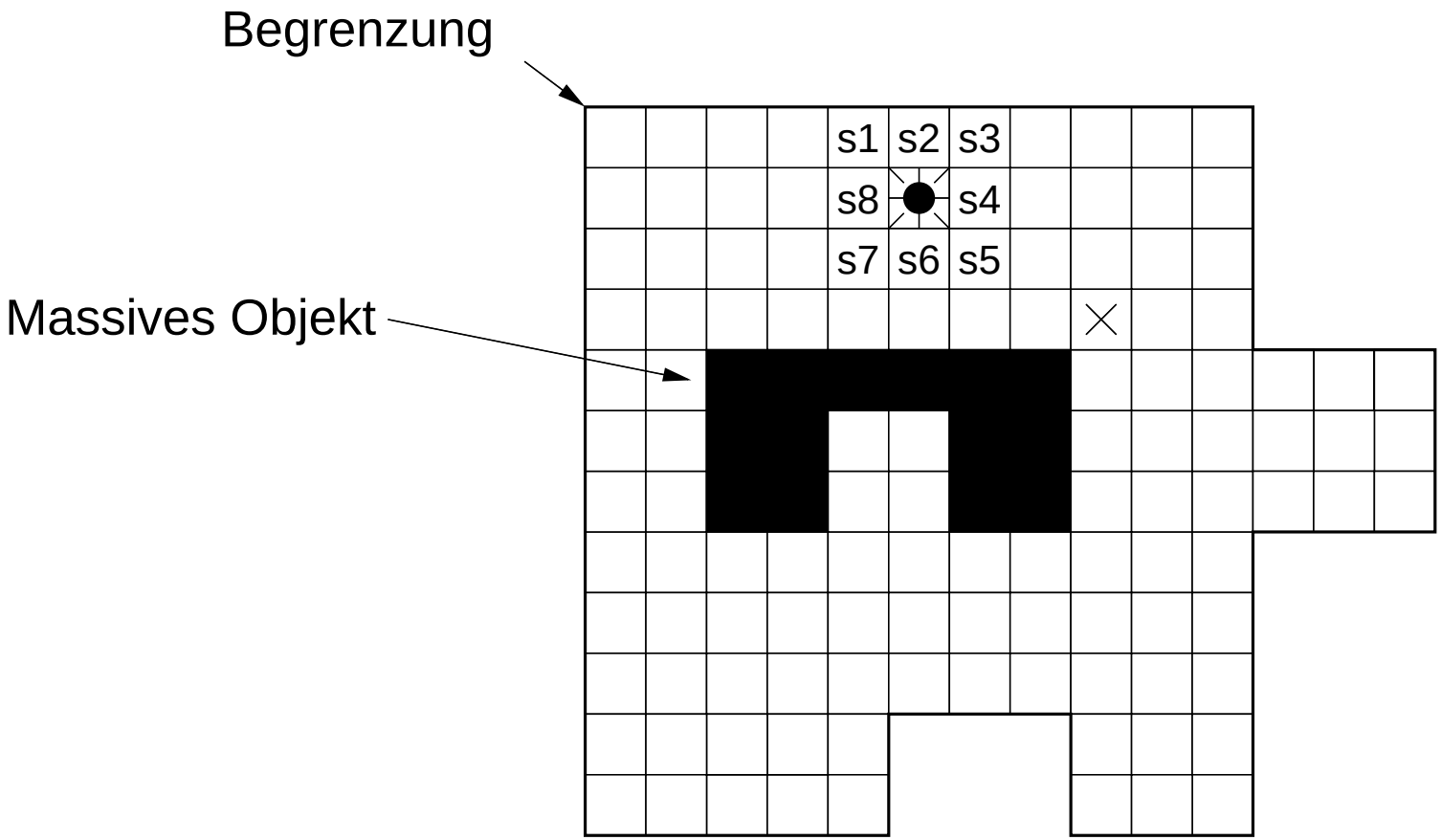
Die Umwelt sei der Einfachheit halber eine fiktive zweidimensionale Gitterzelleneinheit, in der sich verschiedene (Spielzeug-)Agenten tummeln. In den Zellen können Objekte mit verschiedenen Eigenschaften sein, es gibt auch Barrieren. Agenten können von Zelle zu Zelle laufen. Es gibt keine engen Zwischenräume,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





Diese Umgebungen sind *nicht* erlaubt!

Gitterwelt (II)

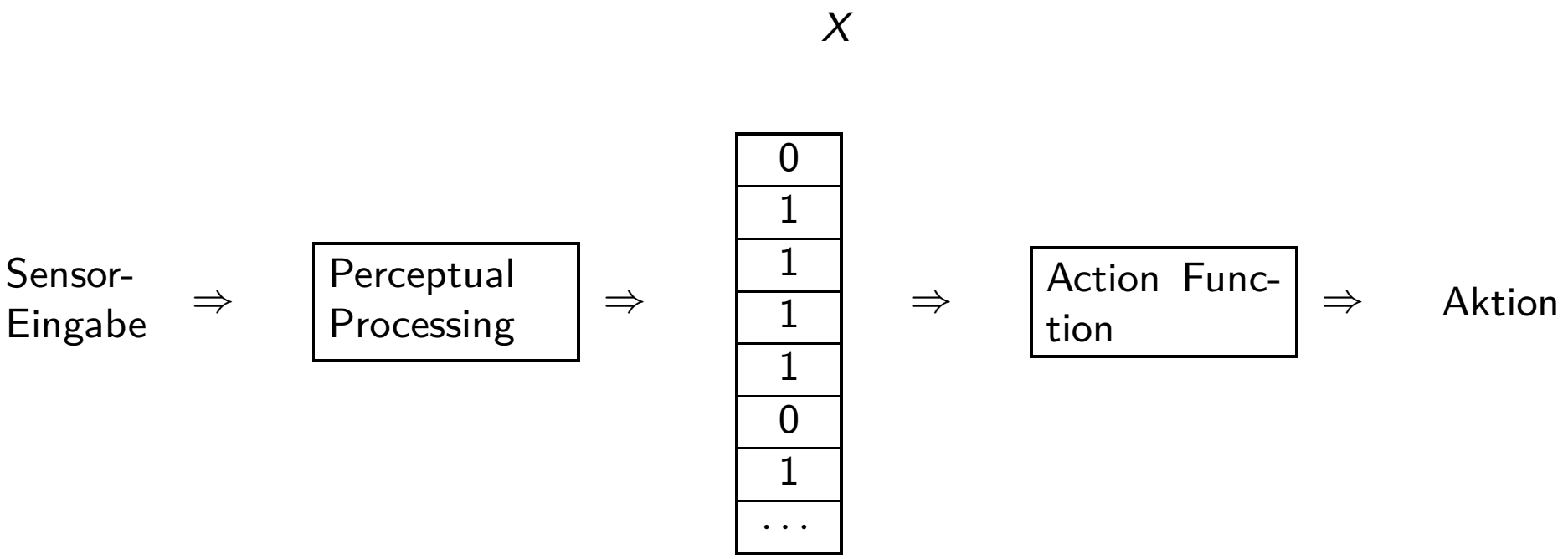


Der Roboter kann mit Hilfe der Sensoren s_1 bis s_8 feststellen, welche Zellen in seiner Nachbarschaft belegt sind.

Gitterwelt (III)

- $s_i \in \{0, 1\}$, Sensoreingabe:
  - $s_j = 0 \Leftrightarrow$ Zelle s_j ist frei für Roboter
  - An der mit $\times$ markierten Stelle: $(0, 0, 0, 0, 0, 0, 1, 0)$
- vier mögliche 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 zwei Schritten gelöst:
  1. Phase: perception processing
  2. Phase: action computation

Komponenten: *Perception* und *Action*



- Eigenschaftsvektor $X = (0, 1, 1, 1, 1, 0, 1, \cdots)^T$
- Vom Entwickler zugewiesene Bedeutungen:
  - $(0, 1, \boxed{\mathbf{1}}, 1, 1, 1, \cdots)$: "an einer Wand"
  - $(0, 1, 1, 1, \boxed{\mathbf{1}}, 1, \cdots)$: "in einer Ecke"

Beispiel: Wandverfolgung (I)

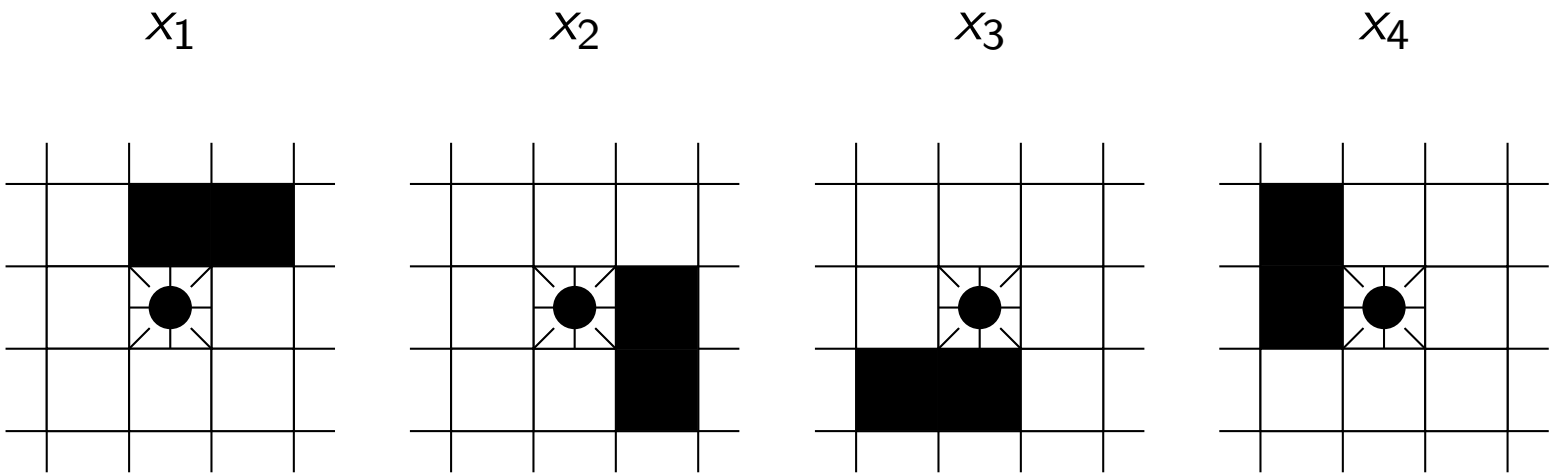
- Aufgabe: Gehe zu einer Zelle an der Begrenzung eines Objekts und folge dieser Grenze

- Perception:
  - Es gibt 2^8 verschiedene Sensoreingaben, von denen einige wegen der Einschränkung *keine engen Zwischenräume* wegfallen.
  - Wir wählen vier Merkmale $x_1, \cdots, x_4$:

$$
\begin{aligned}
x_1 = 1 &\quad\Leftrightarrow\quad (s_2 = 1 \vee s_3 = 1) \\
x_2 = 1 &\quad\Leftrightarrow\quad (s_4 = 1 \vee s_5 = 1) \\
x_3 = 1 &\quad\Leftrightarrow\quad (s_6 = 1 \vee s_7 = 1) \\
x_4 = 1 &\quad\Leftrightarrow\quad (s_8 = 1 \vee s_1 = 1)
\end{aligned}
$$

Beispiel: Wandverfolgung (II)

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



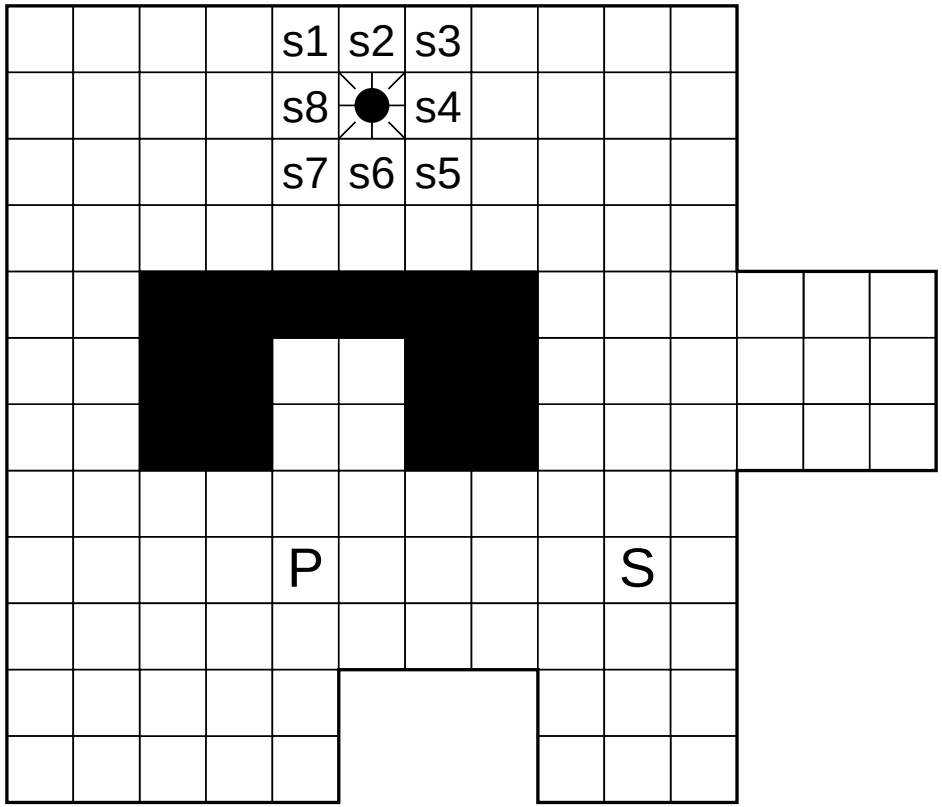
- In komplexen Welten sind Informationen typischerweise unsicher, vage, oder sogar falsch.

Beispiel: Wandverfolgung (III)

- Aktionen
  - Falls keines der vier Merkmale den Wert 1 hat, führe `north` durch.
  - sonst:

$$\text{If } x_1 = 1 \text{ and } x_2 = 0 \text{ then } \texttt{east}$$
$$\text{If } x_2 = 1 \text{ and } x_3 = 0 \text{ then } \texttt{south}$$
$$\text{If } x_3 = 1 \text{ and } x_4 = 0 \text{ then } \texttt{west}$$
$$\text{If } x_4 = 1 \text{ and } x_1 = 0 \text{ then } \texttt{north}$$

Beispiel: Wandverfolgung (IV)



- Ein Roboter, der an Position P startet, bewegt sich entgegen dem Uhrzeigersinn am Objekt entlang.
- Ein Roboter, der an Position S 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 $c_j \longrightarrow a_j$, wobei
  - c_j der Bedingungsteil und
  - a_j der Aktionsteil sind.

- In unserem Beispiel erhält man z.B. die Regeln:

$$\begin{aligned} x_4 \wedge \bar{x_1} &\longrightarrow \text{go north} \\ x_1 \wedge \bar{x_2} &\longrightarrow \text{go east} \\ x_2 \wedge \bar{x_3} &\longrightarrow \text{go south} \\ x_3 \wedge \bar{x_4} &\longrightarrow \text{go west} \\ 1 &\longrightarrow \text{go north} \end{aligned}$$

- 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