

# Künstliche Intelligenz

## Intelligente Agenten

Dr.-Ing. Stefan Lüdtkke

Universität Leipzig

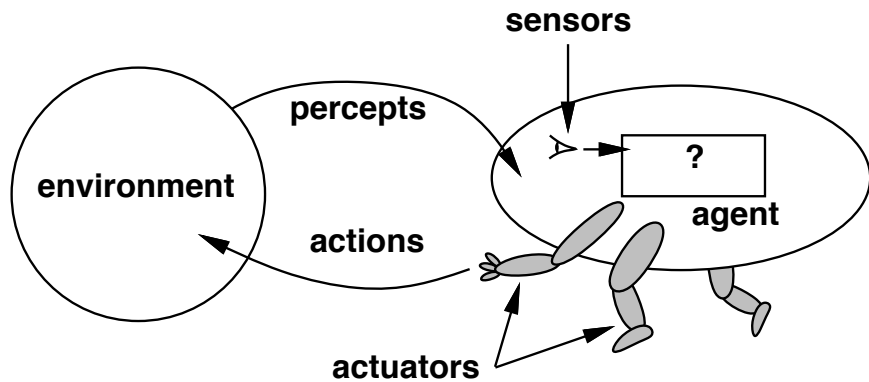
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)

# Agenten

Ein Agent...

- Handelt autonom
- Nimmt seine Umgebung wahr
- Existiert über einen Zeitraum
- Passt sich an Veränderungen an
- Verfolgt Ziele

# Agent und Umgebung



Agenten sind Menschen, Thermostate, Software-Agenten, Roboter, ...

# Agent und Umgebung

## Agentenfunktion und Agentenprogramm

Die Aktion, die ein Agent auswählt, hängt sinnvollerweise nur von den *bisherigen* Beobachtungen ab. Also:

`AgentFunction: [Percept] -> Action`

# Agent und Umgebung

## Agentenfunktion und Agentenprogramm

Die Aktion, die ein Agent auswählt, hängt sinnvollerweise nur von den *bisherigen* Beobachtungen ab. Also:

`AgentFunction: [Percept] -> Action`

Das Argument `[Percept]` ist die Sequenz der bisherigen Beobachtungen.

# Agent und Umgebung

## Agentenfunktion und Agentenprogramm

Die Aktion, die ein Agent auswählt, hängt sinnvollerweise nur von den *bisherigen* Beobachtungen ab. Also:

`AgentFunction: [Percept] -> Action`

Das Argument [Percept] ist die Sequenz der bisherigen Beobachtungen.

- Eine Funktion vom Typ `AgentFunction`, eine *Agentenfunktion*, beschreibt *vollständig* das Verhalten eines Agenten.

# Agent und Umgebung

## Agentenfunktion und Agentenprogramm

Die Aktion, die ein Agent auswählt, hängt sinnvollerweise nur von den *bisherigen* Beobachtungen ab. Also:

`AgentFunction: [Percept] -> Action`

Das Argument `[Percept]` ist die Sequenz der bisherigen Beobachtungen.

- Eine Funktion vom Typ `AgentFunction`, eine *Agentenfunktion*, beschreibt *vollständig* das Verhalten eines Agenten.
- *Jeder* mögliche Agent kann durch eine Funktion vom Typ `AgentFunction` charakterisiert werden.

# Agent und Umgebung

## Agentenfunktion und Agentenprogramm

Die Aktion, die ein Agent auswählt, hängt sinnvollerweise nur von den *bisherigen* Beobachtungen ab. Also:

`AgentFunction: [Percept] -> Action`

Das Argument `[Percept]` ist die Sequenz der bisherigen Beobachtungen.

- Eine Funktion vom Typ `AgentFunction`, eine *Agentenfunktion*, beschreibt *vollständig* das Verhalten eines Agenten.
- *Jeder* mögliche Agent kann durch eine Funktion vom Typ `AgentFunction` charakterisiert werden.
- Die *Agentenfunktion* (ein abstraktes mathematisches Konstrukt) wird durch ein *Agentenprogramm* implementiert, das auf einer physischen Architektur ausgeführt wird.



# Agent und Umgebung

## Agentenfunktion und Agentenprogramm

Die Aktion, die ein Agent auswählt, hängt sinnvollerweise nur von den *bisherigen* Beobachtungen ab. Also:

`AgentFunction: [Percept] -> Action`

Das Argument [Percept] ist die Sequenz der bisherigen Beobachtungen.

- Eine Funktion vom Typ `AgentFunction`, eine *Agentenfunktion*, beschreibt *vollständig* das Verhalten eines Agenten.
- *Jeder* mögliche Agent kann durch eine Funktion vom Typ `AgentFunction` charakterisiert werden.
- Die *Agentenfunktion* (ein abstraktes mathematisches Konstrukt) wird durch ein *Agentenprogramm* implementiert, das auf einer physischen Architektur ausgeführt wird.
- Nicht für jede denkbare *Agentenfunktion* kann ein *Agentenprogramm* gefunden werden. (Berechenbarkeit)

# Nutzen und Rationalität

*Rationales Handeln* ist definiert in Bezug auf ein vorher festgelegtes *Erfolgsmaß*, den *Nutzen*.

# Nutzen und Rationalität

*Rationales Handeln* ist definiert in Bezug auf ein vorher festgelegtes *Erfolgsmaß*, den *Nutzen*. Dieser Nutzen ist definiert als Funktion der Folge von Umgebungszuständen. Also Allgemein:

Utility: [WorldState] -> Double

# Nutzen und Rationalität

*Rationales Handeln* ist definiert in Bezug auf ein vorher festgelegtes *Erfolgmaß*, den *Nutzen*. Dieser Nutzen ist definiert als Funktion der Folge von Umgebungszuständen. Also Allgemein:

Utility: [WorldState] -> Double

Gegeben eine *Folge von Beobachtungen* und ein *Modell des Umgebungsverhaltens* wählt ein *rationaler Agent* diejenige Aktion, die den *erwarteten Nutzen* maximiert.

# Nutzen und Rationalität

*Rationales Handeln* ist definiert in Bezug auf ein vorher festgelegtes *Erfolgmaß*, den *Nutzen*. Dieser Nutzen ist definiert als Funktion der Folge von Umgebungszuständen. Also Allgemein:

Utility: [WorldState]  $\rightarrow$  Double

Gegeben eine *Folge von Beobachtungen* und ein *Modell des Umgebungsverhaltens* wählt ein *rationaler Agent* diejenige Aktion, die den *erwarteten Nutzen* maximiert. Da die Welt nichtdeterministisch auf Aktionen des Agenten reagieren kann, ist es klar, dass wir im Allgemeinen nur einen *Erwartungswert* maximieren können.

# Rationalität

Unterscheidung zwischen Rationalität und *Allwissenheit* sinnvoll:

*Ich laufe die Straße entlang. Auf der anderen Straßenseite sehe ich einen Freund, den ich begrüßen will. Ich schaue nach links und rechts, kein Verkehr zu sehen. Also überquere ich die Straße, und werde von einem herabfallenden Flugzeugteil erschlagen.*

*Sollte in meinem Nachruf stehen, dass ich mich irrational verhalten habe, weil ich die Straße überquert habe?*

# Rationalität

- Rationalität bedeutet *nicht* Allwissenheit

# Rationalität

- Rationalität bedeutet *nicht* Allwissenheit
  - Ein allwissender Agent kennt den tatsächlichen Effekt jeder Aktion.



# Rationalität

- Rationalität bedeutet *nicht* Allwissenheit
  - Ein allwissender Agent kennt den tatsächlichen Effekt jeder Aktion.
  - Beobachtungen liefern möglicherweise nicht alle benötigten Informationen.

# Rationalität

- Rationalität bedeutet *nicht* Allwissenheit
  - Ein allwissender Agent kennt den tatsächlichen Effekt jeder Aktion.
  - Beobachtungen liefern möglicherweise nicht alle benötigten Informationen.
  - Das Modell des Umgebungsverhaltens ist möglicherweise unvollständig oder fehlerhaft.

# Rationalität

- Rationalität bedeutet *nicht* Allwissenheit
  - Ein allwissender Agent kennt den tatsächlichen Effekt jeder Aktion.
  - Beobachtungen liefern möglicherweise nicht alle benötigten Informationen.
  - Das Modell des Umgebungsverhaltens ist möglicherweise unvollständig oder fehlerhaft.
- Rationalität bedeutet *nicht* Hellseherei

# Rationalität

- Rationalität bedeutet *nicht* Allwissenheit
  - Ein allwissender Agent kennt den tatsächlichen Effekt jeder Aktion.
  - Beobachtungen liefern möglicherweise nicht alle benötigten Informationen.
  - Das Modell des Umgebungsverhaltens ist möglicherweise unvollständig oder fehlerhaft.
- Rationalität bedeutet *nicht* Hellseherei
  - Aktionen liefern (in nichtdeterministischen und / oder nur partiell beobachtbaren Umgebungen) nicht sicher das erwartete Ergebnis.

# Rationalität

- Rationalität bedeutet *nicht* Allwissenheit
  - Ein allwissender Agent kennt den tatsächlichen Effekt jeder Aktion.
  - Beobachtungen liefern möglicherweise nicht alle benötigten Informationen.
  - Das Modell des Umgebungsverhaltens ist möglicherweise unvollständig oder fehlerhaft.
- Rationalität bedeutet *nicht* Hellseherei
  - Aktionen liefern (in nichtdeterministischen und / oder nur partiell beobachtbaren Umgebungen) nicht sicher das erwartete Ergebnis.
- (perfekt) *rationales* Handeln heißt also *nicht* notwendig (absolut) *perfektes* Handeln.

# Rationalität

- Rationalität bedeutet *nicht* Allwissenheit
  - Ein allwissender Agent kennt den tatsächlichen Effekt jeder Aktion.
  - Beobachtungen liefern möglicherweise nicht alle benötigten Informationen.
  - Das Modell des Umgebungsverhaltens ist möglicherweise unvollständig oder fehlerhaft.
- Rationalität bedeutet *nicht* Hellseherei
  - Aktionen liefern (in nichtdeterministischen und / oder nur partiell beobachtbaren Umgebungen) nicht sicher das erwartete Ergebnis.
- (perfekt) *rationales* Handeln heißt also *nicht* notwendig (absolut) *perfektes* Handeln.
  - (perfekt) rationales Handeln maximiert den *erwarteten* Nutzen.

# Rationalität

- Rationalität bedeutet *nicht* Allwissenheit
  - Ein allwissender Agent kennt den tatsächlichen Effekt jeder Aktion.
  - Beobachtungen liefern möglicherweise nicht alle benötigten Informationen.
  - Das Modell des Umgebungsverhaltens ist möglicherweise unvollständig oder fehlerhaft.
- Rationalität bedeutet *nicht* Hellseherei
  - Aktionen liefern (in nichtdeterministischen und / oder nur partiell beobachtbaren Umgebungen) nicht sicher das erwartete Ergebnis.
- (perfekt) *rationales* Handeln heißt also *nicht* notwendig (absolut) *perfektes* Handeln.
  - (perfekt) rationales Handeln maximiert den *erwarteten* Nutzen.
  - (absolut) perfektes Handeln maximiert den *tatsächlichen* Nutzen

- Rationalität bedeutet *nicht* Allwissenheit
    - Ein allwissender Agent kennt den tatsächlichen Effekt jeder Aktion.
    - Beobachtungen liefern möglicherweise nicht alle benötigten Informationen.
    - Das Modell des Umgebungsverhaltens ist möglicherweise unvollständig oder fehlerhaft.
  - Rationalität bedeutet *nicht* Hellseherei
    - Aktionen liefern (in nichtdeterministischen und / oder nur partiell beobachtbaren Umgebungen) nicht sicher das erwartete Ergebnis.
  - (perfekt) *rationales* Handeln heißt also *nicht* notwendig (absolut) *perfektes* Handeln.
    - (perfekt) rationales Handeln maximiert den *erwarteten* Nutzen.
    - (absolut) perfektes Handeln maximiert den *tatsächlichen* Nutzen
- ohne Hellseherei lässt sich perfektes Handeln in nichtdeterministischen Umgebungen nicht erreichen.



# Rationalität

## Exploration, Lernen, Autonomie

Gegeben entsprechende Aktionen, kann ein rationaler Agent sich dafür entscheiden, vor kritischen Entscheidungen fehlendes Wissen zu beschaffen:

# Rationalität

## Exploration, Lernen, Autonomie

Gegeben entsprechende Aktionen, kann ein rationaler Agent sich dafür entscheiden, vor kritischen Entscheidungen fehlendes Wissen zu beschaffen:

- *Exploration* der Umgebung

# Rationalität

## Exploration, Lernen, Autonomie

Gegeben entsprechende Aktionen, kann ein rationaler Agent sich dafür entscheiden, vor kritischen Entscheidungen fehlendes Wissen zu beschaffen:

- *Exploration* der Umgebung
- *Lernen* neuer Tatsachen über das Verhalten der Umgebung

# Rationalität

## Exploration, Lernen, Autonomie

Gegeben entsprechende Aktionen, kann ein rationaler Agent sich dafür entscheiden, vor kritischen Entscheidungen fehlendes Wissen zu beschaffen:

- *Exploration* der Umgebung
- *Lernen* neuer Tatsachen über das Verhalten der Umgebung

Die *Autonomie* eines Agenten beschreibt seine Fähigkeit, sein Verhalten im Lauf der Zeit an neuerworbenes Wissen anzupassen.

# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

Beispiel: autonomes Taxi:

# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

Beispiel: autonomes Taxi:

- Performance (Nutzen):
- Environment (Umgebung):
- Actuators (Aktoren, Aktionen):
- Sensors (Percepts, Beobachtungen):

# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

Beispiel: autonomes Taxi:

- Performance (Nutzen):
  - Umsatz, Komfort für Fahrgast, Zeit zum Fahrtziel, Anzahl Strafzettel, Anzahl Unfälle, ...
- Environment (Umgebung):
- Actuators (Aktoren, Aktionen):
- Sensors (Percepts, Beobachtungen):



# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

Beispiel: autonomes Taxi:

- Performance (Nutzen):
  - Umsatz, Komfort für Fahrgast, Zeit zum Fahrtziel, Anzahl Strafzettel, Anzahl Unfälle, ...
- Environment (Umgebung):
  - Straßenverläufe, andere Verkehrsteilnehmer, Fußgänger, Baustellen, Schlaglöcher, Wildwechsel, Fahrgäste, ...
- Actuators (Aktoren, Aktionen):
  
- Sensors (Percepts, Beobachtungen):

# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

Beispiel: autonomes Taxi:

- Performance (Nutzen):
  - Umsatz, Komfort für Fahrgast, Zeit zum Fahrtziel, Anzahl Strafzettel, Anzahl Unfälle, ...
- Environment (Umgebung):
  - Straßenverläufe, andere Verkehrsteilnehmer, Fußgänger, Baustellen, Schlaglöcher, Wildwechsel, Fahrgäste, ...
- Actuators (Aktoren, Aktionen):
  - Bremse, Gas, Kupplung, Gangschaltung, Lenkung, Hupe, Licht, Fensterheber, Klimaanlage, Radio, ...
- Sensors (Percepts, Beobachtungen):

# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

Beispiel: autonomes Taxi:

- Performance (Nutzen):
  - Umsatz, Komfort für Fahrgast, Zeit zum Fahrtziel, Anzahl Strafzettel, Anzahl Unfälle, ...
- Environment (Umgebung):
  - Straßenverläufe, andere Verkehrsteilnehmer, Fußgänger, Baustellen, Schlaglöcher, Wildwechsel, Fahrgäste, ...
- Actuators (Aktoren, Aktionen):
  - Bremse, Gas, Kupplung, Gangschaltung, Lenkung, Hupe, Licht, Fensterheber, Klimaanlage, Radio, ...
- Sensors (Percepts, Beobachtungen):
  - Kameras, Laser-Entfernungsmesser, GPS, Radar, Akzelerometer, Öltemperatur, Wassertemperatur, Tankanzeiger, Innenraumüberwachung, Tastatur, Mikrofon, ...

# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

Beispiel: Internet-Shopping-Agent

- Performance (Nutzen):
- Environment (Umgebung):
- Actuators (Aktoren, Aktionen):
- Sensors (Percepts, Beobachtungen):

# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

Beispiel: Internet-Shopping-Agent

- Performance (Nutzen):
  - Preis, Qualität, Nützlichkeit, Lieferzeitpunkt, ...
- Environment (Umgebung):
- Actuators (Aktoren, Aktionen):
- Sensors (Percepts, Beobachtungen):

# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

Beispiel: Internet-Shopping-Agent

- Performance (Nutzen):
  - Preis, Qualität, Nützlichkeit, Lieferzeitpunkt, ...
- Environment (Umgebung):
  - Web-Shops
- Actuators (Aktoren, Aktionen):
- Sensors (Percepts, Beobachtungen):

# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

Beispiel: Internet-Shopping-Agent

- Performance (Nutzen):
  - Preis, Qualität, Nützlichkeit, Lieferzeitpunkt, ...
- Environment (Umgebung):
  - Web-Shops
- Actuators (Aktoren, Aktionen):
  - HTTP-Kommandos; Interaktion mit HTML-Seite
- Sensors (Percepts, Beobachtungen):

# Die Aufgaben-Umgebung

PEAS: Performance, Environment, Actuators, Sensors

Für die Entwicklung eines rationalen Agenten muss die Aufgaben-Umgebung festgelegt werden.

Beispiel: Internet-Shopping-Agent

- Performance (Nutzen):
  - Preis, Qualität, Nützlichkeit, Lieferzeitpunkt, ...
- Environment (Umgebung):
  - Web-Shops
- Actuators (Aktoren, Aktionen):
  - HTTP-Kommandos; Interaktion mit HTML-Seite
- Sensors (Percepts, Beobachtungen):
  - HTML-Struktur (Text, Grafik, Skripte)



# Umgebungstypen

Umgebungsattribute: „einfach“ vs „schwierig“

# Umgebungstypen

Umgebungsattribute: „einfach“ vs „schwierig“

- Vollständige vs. partielle Observierbarkeit?

# Umgebungstypen

Umgebungsattribute: „einfach“ vs „schwierig“

- Vollständige vs. partielle Observierbarkeit?
- Deterministische vs. nicht-deterministische Reaktion auf Aktionen?

# Umgebungstypen

Umgebungsattribute: „einfach“ vs „schwierig“

- Vollständige vs. partielle Observierbarkeit?
- Deterministische vs. nicht-deterministische Reaktion auf Aktionen?
  - Wird der Nichtdeterminismus nur durch die Handlung anderer Agenten erzeugt, ist die Umgebung *strategisch*.

# Umgebungstypen

Umgebungsattribute: „einfach“ vs „schwierig“

- Vollständige vs. partielle Observierbarkeit?
- Deterministische vs. nicht-deterministische Reaktion auf Aktionen?
  - Wird der Nichtdeterminismus nur durch die Handlung anderer Agenten erzeugt, ist die Umgebung *strategisch*.
- Episodische vs. sequentielle Aufgabenstellungen?

Kann jede Beobachtung für sich allein bearbeitet werden oder hängt die Reaktion auf eine Beobachtung auch von vorherigen Aktivitäten ab?

# Umgebungstypen

Umgebungsattribute: „einfach“ vs „schwierig“

- Vollständige vs. partielle Observierbarkeit?
- Deterministische vs. nicht-deterministische Reaktion auf Aktionen?
  - Wird der Nichtdeterminismus nur durch die Handlung anderer Agenten erzeugt, ist die Umgebung *strategisch*.
- Episodische vs. sequentielle Aufgabenstellungen?

Kann jede Beobachtung für sich allein bearbeitet werden oder hängt die Reaktion auf eine Beobachtung auch von vorherigen Aktivitäten ab?
- Statische vs. dynamische Umgebung?

Kann sich während des Aktionsauswahlprozesses die Umgebung verändern?

# Umgebungstypen

Umgebungsattribute: „einfach“ vs „schwierig“

- Vollständige vs. partielle Observierbarkeit?
- Deterministische vs. nicht-deterministische Reaktion auf Aktionen?
  - Wird der Nichtdeterminismus nur durch die Handlung anderer Agenten erzeugt, ist die Umgebung *strategisch*.
- Episodische vs. sequentielle Aufgabenstellungen?

Kann jede Beobachtung für sich allein bearbeitet werden oder hängt die Reaktion auf eine Beobachtung auch von vorherigen Aktivitäten ab?
- Statische vs. dynamische Umgebung?

Kann sich während des Aktionsauswahlprozesses die Umgebung verändern?
- Diskrete oder kontinuierliche Zustands- / Beobachtungsräume?

# Umgebungstypen

Umgebungsattribute: „einfach“ vs „schwierig“

- Vollständige vs. partielle Observierbarkeit?
- Deterministische vs. nicht-deterministische Reaktion auf Aktionen?
  - Wird der Nichtdeterminismus nur durch die Handlung anderer Agenten erzeugt, ist die Umgebung *strategisch*.
- Episodische vs. sequentielle Aufgabenstellungen?  
Kann jede Beobachtung für sich allein bearbeitet werden oder hängt die Reaktion auf eine Beobachtung auch von vorherigen Aktivitäten ab?
- Statische vs. dynamische Umgebung?  
Kann sich während des Aktionsauswahlprozesses die Umgebung verändern?
- Diskrete oder kontinuierliche Zustands- / Beobachtungsräume?
- Single-Agent vs. Multi-Agent-Situation?  
Gibt es Mitspieler oder nur eine “indifferente” Umgebung?  
Kooperieren Mitspieler für eine Lösung, oder stehen sie im Wettbewerb?



# Umgebungstypen

## Single-Agent vs. Multi-Agent

Was ist für einen Agenten  $A$  der Unterschied zwischen einer indifferenten nichtdeterministischen Umgebung und einem anderen Agenten  $B$ ?

# Umgebungstypen

## Single-Agent vs. Multi-Agent

Was ist für einen Agenten  $A$  der Unterschied zwischen einer indifferenten nichtdeterministischen Umgebung und einem anderen Agenten  $B$ ?

- Ein anderer Agent wird potentiell versuchen, ein Leistungsmaß zu optimieren. Dabei kann das Leistungsmaß von  $B$  vom Leistungsmaß für  $A$  abhängen:
  - Kooperation: positiv korreliert
  - Wettbewerb: negativ korreliert

# Umgebungstypen

## Single-Agent vs. Multi-Agent

Was ist für einen Agenten  $A$  der Unterschied zwischen einer indifferenten nichtdeterministischen Umgebung und einem anderen Agenten  $B$ ?

- Ein anderer Agent wird potentiell versuchen, ein Leistungsmaß zu optimieren. Dabei kann das Leistungsmaß von  $B$  vom Leistungsmaß für  $A$  abhängen:
  - Kooperation: positiv korreliert
  - Wettbewerb: negativ korreliert
- D.h., es liegt möglicherweise zusätzliches Wissen über die Reaktion der Umgebung vor, das  $A$  für die Aktionsauswahl verwenden kann.

# Umgebungstypen

## Single-Agent vs. Multi-Agent

Was ist für einen Agenten  $A$  der Unterschied zwischen einer indifferenten nichtdeterministischen Umgebung und einem anderen Agenten  $B$ ?

- Ein anderer Agent wird potentiell versuchen, ein Leistungsmaß zu optimieren. Dabei kann das Leistungsmaß von  $B$  vom Leistungsmaß für  $A$  abhängen:
  - Kooperation: positiv korreliert
  - Wettbewerb: negativ korreliert
- D.h., es liegt möglicherweise zusätzliches Wissen über die Reaktion der Umgebung vor, das  $A$  für die Aktionsauswahl verwenden kann.
- Die *Kommunikation* zwischen Agenten (zum Zweck der Kooperation) kann sinnvoll sein.

# Umgebungstypen

## Single-Agent vs. Multi-Agent

Was ist für einen Agenten  $A$  der Unterschied zwischen einer indifferenten nichtdeterministischen Umgebung und einem anderen Agenten  $B$ ?

- Ein anderer Agent wird potentiell versuchen, ein Leistungsmaß zu optimieren. Dabei kann das Leistungsmaß von  $B$  vom Leistungsmaß für  $A$  abhängen:
  - Kooperation: positiv korreliert
  - Wettbewerb: negativ korreliert
- D.h., es liegt möglicherweise zusätzliches Wissen über die Reaktion der Umgebung vor, das  $A$  für die Aktionsauswahl verwenden kann.
- Die *Kommunikation* zwischen Agenten (zum Zweck der Kooperation) kann sinnvoll sein.
- Absichtlich *nichtdeterministisches* Verhalten von  $A$  kann sinnvoll sein (im Fall von Wettbewerbssituationen).

# Umgebungstypen

## Einige Beispiele

	Solitär	Backgammon	Internet-Shopping	Taxi
Observabel?				

# Umgebungstypen

## Einige Beispiele

	Solitär	Backgammon	Internet-Shopping	Taxi
Observabel?	ja	ja	nein	nein
Deterministisch?				

# Umgebungstypen

## Einige Beispiele

	Solitär	Backgammon	Internet-Shopping	Taxi
Observabel?	ja	ja	nein	nein
Deterministisch?	ja	nein	partiell	nein
Episodisch?				



# Umgebungstypen

## Einige Beispiele

	Solitär	Backgammon	Internet-Shopping	Taxi
Observabel?	ja	ja	nein	nein
Deterministisch?	ja	nein	partiell	nein
Episodisch?	nein	nein	nein	nein
Statisch				

# Umgebungstypen

## Einige Beispiele

	Solitär	Backgammon	Internet-Shopping	Taxi
Observabel?	ja	ja	nein	nein
Deterministisch?	ja	nein	partiell	nein
Episodisch?	nein	nein	nein	nein
Statisch	ja	ja	nein	nein
Diskret?				

# Umgebungstypen

## Einige Beispiele

	Solitär	Backgammon	Internet-Shopping	Taxi
Observabel?	ja	ja	nein	nein
Deterministisch?	ja	nein	partiell	nein
Episodisch?	nein	nein	nein	nein
Statisch	ja	ja	nein	nein
Diskret?	ja	ja	ja	nein
Single-Agent?				

# Umgebungstypen

## Einige Beispiele

	Solitär	Backgammon	Internet-Shopping	Taxi
Observabel?	ja	ja	nein	nein
Deterministisch?	ja	nein	partiell	nein
Episodisch?	nein	nein	nein	nein
Statisch	ja	ja	nein	nein
Diskret?	ja	ja	ja	nein
Single-Agent?	ja	nein	ja	nein

# Umgebungstypen

## Einige Beispiele

	Solitär	Backgammon	Internet-Shopping	Taxi
Observabel?	ja	ja	nein	nein
Deterministisch?	ja	nein	partiell	nein
Episodisch?	nein	nein	nein	nein
Statisch	ja	ja	nein	nein
Diskret?	ja	ja	ja	nein
Single-Agent?	ja	nein	ja	nein

- Der Umgebungstyp bestimmt weitgehend den Entwurf des Agenten

# Umgebungstypen

## Einige Beispiele

	Solitär	Backgammon	Internet-Shopping	Taxi
Observabel?	ja	ja	nein	nein
Deterministisch?	ja	nein	partiell	nein
Episodisch?	nein	nein	nein	nein
Statisch	ja	ja	nein	nein
Diskret?	ja	ja	ja	nein
Single-Agent?	ja	nein	ja	nein

- Der Umgebungstyp bestimmt weitgehend den Entwurf des Agenten
- Viele deterministische Systeme sind so kompliziert, dass man sie als nicht-deterministische-Systeme behandeln muss.

# Umgebungstypen

## Einige Beispiele

	Solitär	Backgammon	Internet-Shopping	Taxi
Observabel?	ja	ja	nein	nein
Deterministisch?	ja	nein	partiell	nein
Episodisch?	nein	nein	nein	nein
Statisch	ja	ja	nein	nein
Diskret?	ja	ja	ja	nein
Single-Agent?	ja	nein	ja	nein

- Der Umgebungstyp bestimmt weitgehend den Entwurf des Agenten
- Viele deterministische Systeme sind so kompliziert, dass man sie als nicht-deterministische-Systeme behandeln muss.
- Die reale Welt ist selbstverständlich: nur partiell observierbar, nichtdeterministisch, sequentiell, dynamisch, kontinuierlich, multi-agent.

# Agententypen

Eine Vielzahl von Agentenarchitekturen sind denkbar; wir schauen uns exemplarisch vier Varianten wachsender Komplexität an:

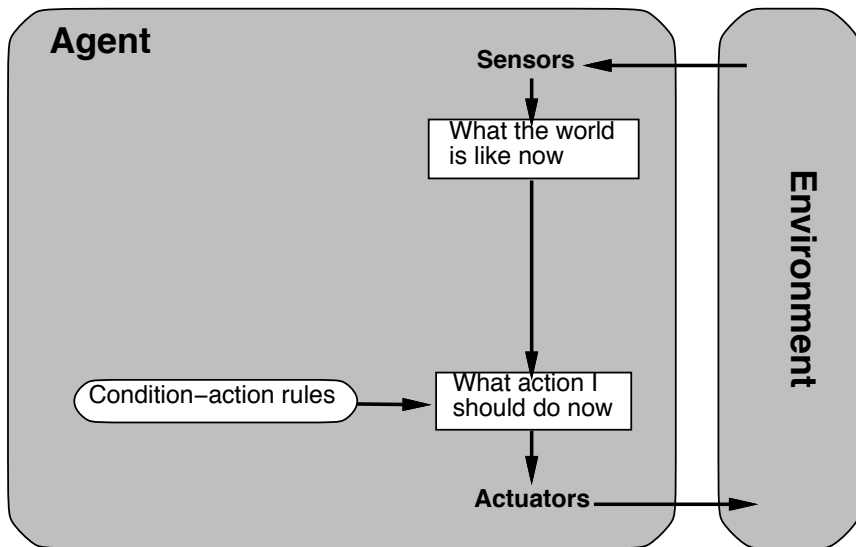
- Einfache Reflex-Agenten
- Zustandsbehaftete Reflex-Agenten  
(modellbasierte Reflex-Agenten)
- Zielbasierte Agenten
- Nutzenbasierte Agenten

(Alle diese grundlegenden Architekturen können auch zu lernfähigen Systemen erweitert werden.)



# Reflex-Agenten

## Aufbau



# Reflex-Agenten

## Diskussion

Reflexagenten funktionieren nur, wenn die Aktion *eindeutig* auf Basis der aktuellen Beobachtung gewählt werden kann. D.h., wenn die Umgebung – in Bezug auf den Handlungsspielraum des Agenten – vollständig beobachtbar ist.

# Reflex-Agenten

## Diskussion

Reflexagenten funktionieren nur, wenn die Aktion *eindeutig* auf Basis der aktuellen Beobachtung gewählt werden kann. D.h., wenn die Umgebung – in Bezug auf den Handlungsspielraum des Agenten – vollständig beobachtbar ist.

Die Lösung ist, wir geben dem Agenten ein *Modell* von der Entwicklung der Teile der Welt, die er nicht beobachten kann.

# Reflex-Agenten

## Diskussion

Reflexagenten funktionieren nur, wenn die Aktion *eindeutig* auf Basis der aktuellen Beobachtung gewählt werden kann. D.h., wenn die Umgebung – in Bezug auf den Handlungsspielraum des Agenten – vollständig beobachtbar ist.

Die Lösung ist, wir geben dem Agenten ein *Modell* von der Entwicklung der Teile der Welt, die er nicht beobachten kann. Ein solches Modell berücksichtigt im Allgemeinen sowohl die Wirkung der Aktionen des Agenten, wie auch die Eigendynamik der Umgebung.

# Reflex-Agenten

## Diskussion

Reflexagenten funktionieren nur, wenn die Aktion *eindeutig* auf Basis der aktuellen Beobachtung gewählt werden kann. D.h., wenn die Umgebung – in Bezug auf den Handlungsspielraum des Agenten – vollständig beobachtbar ist.

Die Lösung ist, wir geben dem Agenten ein *Modell* von der Entwicklung der Teile der Welt, die er nicht beobachten kann. Ein solches Modell berücksichtigt im Allgemeinen sowohl die Wirkung der Aktionen des Agenten, wie auch die Eigendynamik der Umgebung. Für die Teile der Welt, die *nicht* von der aktuellen Beobachtung geliefert werden, muss der Agent ein internes Bild verwalten und ständig Aktualisieren – dies ist der *Zustand*.

# Reflex-Agenten

## Diskussion

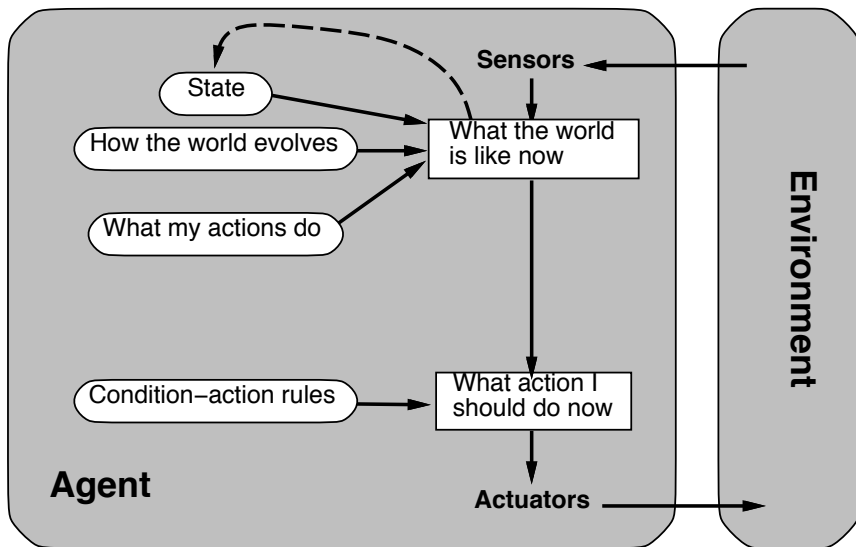
Reflexagenten funktionieren nur, wenn die Aktion *eindeutig* auf Basis der aktuellen Beobachtung gewählt werden kann. D.h., wenn die Umgebung – in Bezug auf den Handlungsspielraum des Agenten – vollständig beobachtbar ist.

Die Lösung ist, wir geben dem Agenten ein *Modell* von der Entwicklung der Teile der Welt, die er nicht beobachten kann. Ein solches Modell berücksichtigt im Allgemeinen sowohl die Wirkung der Aktionen des Agenten, wie auch die Eigendynamik der Umgebung. Für die Teile der Welt, die *nicht* von der aktuellen Beobachtung geliefert werden, muss der Agent ein internes Bild verwalten und ständig Aktualisieren – dies ist der *Zustand*.

Um genau zu sein sollte man zwischen dem *wahren* Zustand der Welt und dem *Bild*, das der Agent von diesem Zustand hat, unterscheiden.

# Zustandsbehaftete Reflex-Agenten

## Aufbau



# Zustandsbehaftete Reflex-Agenten

## Diskussion

- Gegeben einen Zustand und eine Beobachtung wählt ein zustandsbehafteter Reflex-Agent stets dieselbe Aktion.



# Zustandsbehaftete Reflex-Agenten

## Diskussion

- Gegeben einen Zustand und eine Beobachtung wählt ein zustandsbehafteter Reflex-Agent stets dieselbe Aktion.
- Jedoch sind Zustand und Beobachtung nicht immer ausreichend für die Aktionsauswahl.

# Zustandsbehaftete Reflex-Agenten

## Diskussion

- Gegeben einen Zustand und eine Beobachtung wählt ein zustandsbehafteter Reflex-Agent stets dieselbe Aktion.
- Jedoch sind Zustand und Beobachtung nicht immer ausreichend für die Aktionsauswahl. In welche Richtung ein autonomes Taxi an einer Kreuzung abbiegt hängt nicht nur vom Zustand (Position des Fahrzeugs) und der Beobachtung (Kreuzung frei) ab, sondern auch vom *Ziel* des Fahrgastes.

# Zustandsbehaftete Reflex-Agenten

## Diskussion

- Gegeben einen Zustand und eine Beobachtung wählt ein zustandsbehafteter Reflex-Agent stets dieselbe Aktion.
- Jedoch sind Zustand und Beobachtung nicht immer ausreichend für die Aktionsauswahl. In welche Richtung ein autonomes Taxi an einer Kreuzung abbiegt hängt nicht nur vom Zustand (Position des Fahrzeugs) und der Beobachtung (Kreuzung frei) ab, sondern auch vom *Ziel* des Fahrgastes.
- In diesem Fall muss ein Agent auch in der Lage sein zu bewerten, inwieweit eine der möglichen Aktionen ihn dabei unterstützt, sein aktuelles Ziel zu erreichen.

# Zustandsbehaftete Reflex-Agenten

## Diskussion

- Gegeben einen Zustand und eine Beobachtung wählt ein zustandsbehafteter Reflex-Agent stets dieselbe Aktion.
- Jedoch sind Zustand und Beobachtung nicht immer ausreichend für die Aktionsauswahl. In welche Richtung ein autonomes Taxi an einer Kreuzung abbiegt hängt nicht nur vom Zustand (Position des Fahrzeugs) und der Beobachtung (Kreuzung frei) ab, sondern auch vom *Ziel* des Fahrgastes.
- In diesem Fall muss ein Agent auch in der Lage sein zu bewerten, inwieweit eine der möglichen Aktionen ihn dabei unterstützt, sein aktuelles Ziel zu erreichen.

Dazu muss er extrapolieren, wie seine Aktionen die Welt verändern, und bewerten, ob ihn diese veränderte Welt seinem Ziel näher bringt.

# Zielbasierte Agenten

## Konzept

- Ein zielbasierter Agent ist in der Lage, die Wirkung seiner Aktionen einzuschätzen *bevor er diese real ausführt* – er kann somit über sein zukünftiges Handeln spekulieren.

# Zielbasierte Agenten

## Konzept

- Ein zielbasierter Agent ist in der Lage, die Wirkung seiner Aktionen einzuschätzen *bevor er diese real ausführt* – er kann somit über sein zukünftiges Handeln spekulieren.
- Techniken der *Suche* und *Planung* werden genutzt, um für einen gegebenen Anfangszustand, einen gewünschten Zielzustand und eine Menge von Aktionen eine Aktionsfolge zu finden, die vom Ausgangszustand zum Ziel führt.

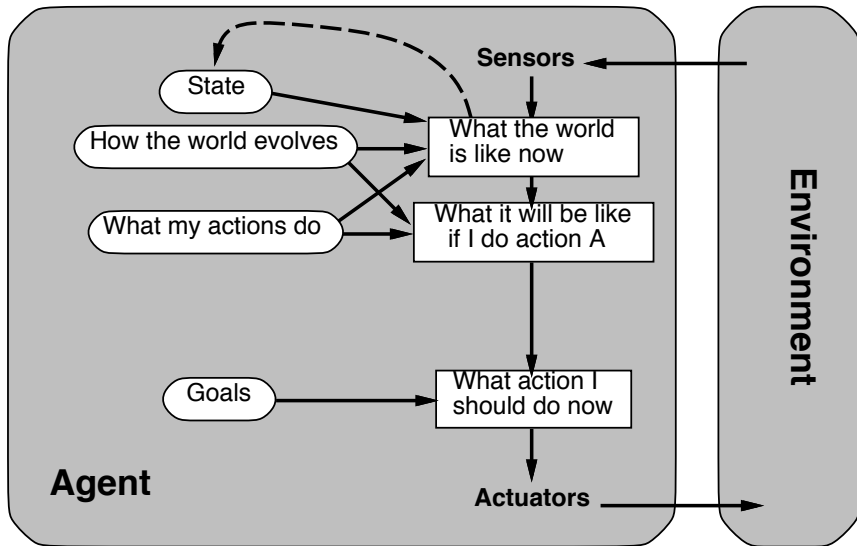
# Zielbasierte Agenten

## Konzept

- Ein zielbasierter Agent ist in der Lage, die Wirkung seiner Aktionen einzuschätzen *bevor er diese real ausführt* – er kann somit über sein zukünftiges Handeln spekulieren.
- Techniken der *Suche* und *Planung* werden genutzt, um für einen gegebenen Anfangszustand, einen gewünschten Zielzustand und eine Menge von Aktionen eine Aktionsfolge zu finden, die vom Ausgangszustand zum Ziel führt.
- Für reflexbasierte Agenten muss vom Designer vorgegeben werden, in welcher Situation wie reagiert werden muss. Ein zielbasierter Agent kann diese Analyse selbständig, bei Bedarf durchführen.

# Zielbasierte Agenten

## Aufbau





# Zielbasierte Agenten

## Diskussion

- Gegeben ein Ziel kann es sein, dass es unterschiedliche Pläne gibt, die alle dieses Ziel erreichen.

# Zielbasierte Agenten

## Diskussion

- Gegeben ein Ziel kann es sein, dass es unterschiedliche Pläne gibt, die alle dieses Ziel erreichen.
- Welchen Plan soll der Agent auswählen? Die schnellste Route? Die schönste Route? Die billigste?

# Zielbasierte Agenten

## Diskussion

- Gegeben ein Ziel kann es sein, dass es unterschiedliche Pläne gibt, die alle dieses Ziel erreichen.
- Welchen Plan soll der Agent auswählen? Die schnellste Route? Die schönste Route? Die billigste?
- Hierzu benötigt er ein Kriterium, das über den *Nutzen* eines Plans Auskunft gibt.  
(Ein rein zielbasierter Agent hat ein grobes binäres Kriterium: „Ziel erreicht: ja/nein?“)

# Zielbasierte Agenten

## Diskussion

- Gegeben ein Ziel kann es sein, dass es unterschiedliche Pläne gibt, die alle dieses Ziel erreichen.
- Welchen Plan soll der Agent auswählen? Die schnellste Route? Die schönste Route? Die billigste?
- Hierzu benötigt er ein Kriterium, das über den *Nutzen* eines Plans Auskunft gibt.  
(Ein rein zielbasierter Agent hat ein grobes binäres Kriterium: „Ziel erreicht: ja/nein?“)
- Dieses Kriterium ist die Nutzen-Funktion, eine Funktion  
`Utility: [State] -> Double`

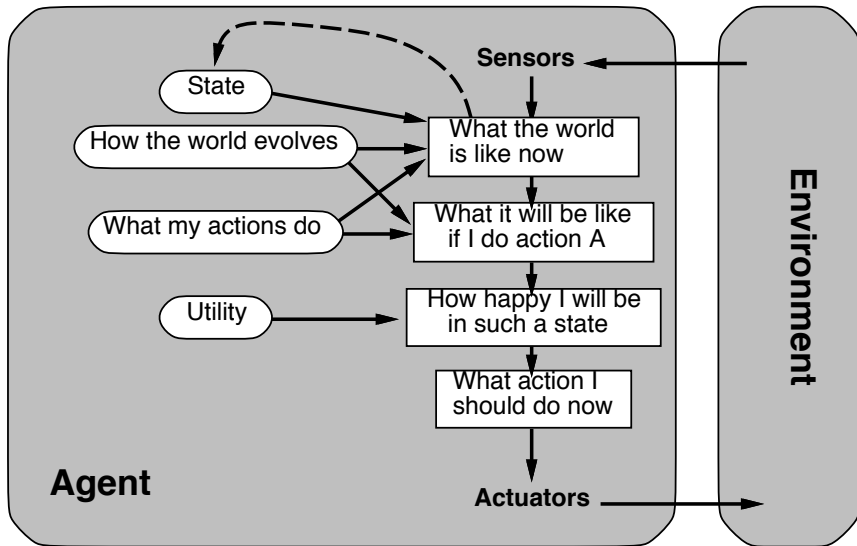
# Zielbasierte Agenten

## Diskussion

- Gegeben ein Ziel kann es sein, dass es unterschiedliche Pläne gibt, die alle dieses Ziel erreichen.
- Welchen Plan soll der Agent auswählen? Die schnellste Route? Die schönste Route? Die billigste?
- Hierzu benötigt er ein Kriterium, das über den *Nutzen* eines Plans Auskunft gibt.  
(Ein rein zielbasierter Agent hat ein grobes binäres Kriterium: „Ziel erreicht: ja/nein?“)
- Dieses Kriterium ist die Nutzen-Funktion, eine Funktion  
`Utility: [State] -> Double`
- Auf dieser Basis kann ein Agent nun zwischen verschiedenen möglichen Plänen wählen, oder auch zwischen verschiedenen, zueinander in Konflikt stehenden Zielen auswählen.

# Nutzenbasierte Agenten

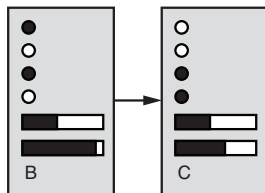
## Aufbau



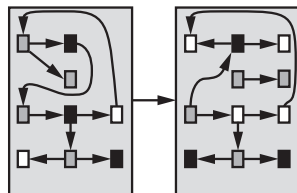
# Was ist ein Zustand?



(a) Atomic



(b) Factored

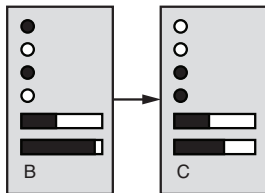


(b) Structured

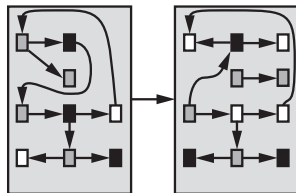
# Was ist ein Zustand?



(a) Atomic



(b) Factored



(b) Structured

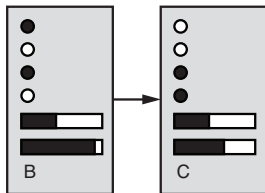
- 1** Atomare Repräsentation. Zustände (B, C) sind Black Boxes ohne interne Struktur



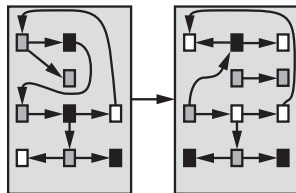
# Was ist ein Zustand?



(a) Atomic



(b) Factored



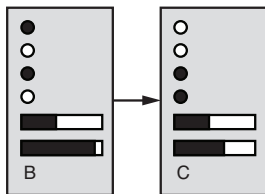
(b) Structured

- 1 Atomare Repräsentation. Zustände (B, C) sind Black Boxes ohne interne Struktur
- 2 Faktorierte Repräsentation. Zustände sind Abbildungen von Attributen auf Werte; Werte können Wahrheitswerte, Symbole, Fließkommazahlen sein. (Beispielsweise realisiert als C struct)

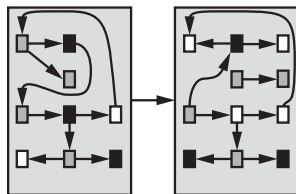
# Was ist ein Zustand?



(a) Atomic



(b) Factored



(b) Structured

- 1 Atomare Repräsentation. Zustände (B, C) sind Black Boxes ohne interne Struktur
- 2 Faktorierte Repräsentation. Zustände sind Abbildungen von Attributen auf Werte; Werte können Wahrheitswerte, Symbole, Fließkommazahlen sein. (Beispielsweise realisiert als C struct)
- 3 Zustände sind beliebige programmiersprachliche Datenstrukturen. (Beispielsweise als CLOS, C++, Java Objektstruktur)

# Zusammenfassung

- *Agenten* interagieren mit ihrer *Umgebung* über *Sensoren* und *Aktoren*

# Zusammenfassung

- *Agenten* interagieren mit ihrer *Umgebung* über *Sensoren* und *Aktoren*
- Die *Agentenfunktion* (`AgentFunction`) ist eine *vollständige* Beschreibung des Agentenverhaltens

# Zusammenfassung

- *Agenten* interagieren mit ihrer *Umgebung* über *Sensoren* und *Aktoren*
- Die *Agentenfunktion* (`AgentFunction`) ist eine *vollständige* Beschreibung des Agentenverhaltens
- Die *Nutzenfunktion* bewertet die Folge von Umgebungszuständen

# Zusammenfassung

- *Agenten* interagieren mit ihrer *Umgebung* über *Sensoren* und *Aktoren*
- Die *Agentenfunktion* (`AgentFunction`) ist eine *vollständige* Beschreibung des Agentenverhaltens
- Die *Nutzenfunktion* bewertet die Folge von Umgebungszuständen
- Ein *perfekt rationaler Agent* maximiert den *erwarteten Nutzen*

# Zusammenfassung

- *Agenten* interagieren mit ihrer *Umgebung* über *Sensoren* und *Aktoren*
- Die *Agentenfunktion* (`AgentFunction`) ist eine *vollständige* Beschreibung des Agentenverhaltens
- Die *Nutzenfunktion* bewertet die Folge von Umgebungszuständen
- Ein *perfekt rationaler Agent* maximiert den *erwarteten Nutzen*
- *Agentenprogramme* implementieren (einige) Agentenfunktionen

# Zusammenfassung

- *Agenten* interagieren mit ihrer *Umgebung* über *Sensoren* und *Aktoren*
- Die *Agentenfunktion* (`AgentFunction`) ist eine *vollständige* Beschreibung des Agentenverhaltens
- Die *Nutzenfunktion* bewertet die Folge von Umgebungszuständen
- Ein *perfekt rationaler Agent* maximiert den *erwarteten Nutzen*
- *Agentenprogramme* implementieren (einige) Agentenfunktionen
- *PEAS*-Beschreibungen definieren die Aufgaben-Umgebung eines Agenten



# Zusammenfassung

- *Agenten* interagieren mit ihrer *Umgebung* über *Sensoren* und *Aktoren*
- Die *Agentenfunktion* (`AgentFunction`) ist eine *vollständige* Beschreibung des Agentenverhaltens
- Die *Nutzenfunktion* bewertet die Folge von Umgebungszuständen
- Ein *perfekt rationaler Agent* maximiert den *erwarteten Nutzen*
- *Agentenprogramme* implementieren (einige) Agentenfunktionen
- *PEAS*-Beschreibungen definieren die Aufgaben-Umgebung eines Agenten
- Umgebungen können in Bezug auf verschiedene Attribute kategorisiert werden:
  - beobachtbar? deterministisch? episodisch? statisch? diskret?
  - Single-Agent?

# Zusammenfassung

- *Agenten* interagieren mit ihrer *Umgebung* über *Sensoren* und *Aktoren*
- Die *Agentenfunktion* (`AgentFunction`) ist eine *vollständige* Beschreibung des Agentenverhaltens
- Die *Nutzenfunktion* bewertet die Folge von Umgebungszuständen
- Ein *perfekt rationaler Agent* maximiert den *erwarteten Nutzen*
- *Agentenprogramme* implementieren (einige) Agentenfunktionen
- *PEAS*-Beschreibungen definieren die Aufgaben-Umgebung eines Agenten
- Umgebungen können in Bezug auf verschiedene Attribute kategorisiert werden:
  - beobachtbar? deterministisch? episodisch? statisch? diskret?  
Single-Agent?
- Es gibt eine Reihe grundlegender Agenten-Architekturen:
  - Reflex, zustandsbehafteter Reflex, zielbasiert, nutzenbasiert