Reasoning and decision-making under uncertainty

9. Vorlesung

Actions, interventions and complex decisions

Sebastian Ptock
AG Sociable Agents

Rückblick: Decision-Making

A decision leads to states with values, or situations with further options for decisions with uncertain outcomes (uncertain values).

Rückblick: Decision-Making

Rationales Decision-Making hängt ab von:

- der **Wahrscheinlichkeit**, Ziele zu einem nötigen Grad zu erreichen
- der **subjektiven Relevanz** von Zielen, modelliert als Präferenzen möglicher Ergebnisse (*risks*, *costs*, *rewards*, etc.)

Decision theory = Probability theory + Utility theory

Given preferences satisfying the constraints
there exists a real-valued function U such that

$$U(A) \geq U(B) \quad \Leftrightarrow \quad A \succsim B$$

$$U([p_1, S_1; \ \ldots \ ; \ p_n, S_n]) = \Sigma_i \ p_i U(S_i)$$

Rückblick: Decision-Making

Prinzip der Maximum Expected Utility (MEU)

Ein Agent ist rational dann und nur dann, wenn er die Aktion wählt, welche die höchste *expected utility* einbringt, gemittelt über alle möglichen Ergebnisse der Aktion.

Anmerkung: eine nicht-deterministische Aktion kann verschiedene Ergebnisse $Result_i(A)$ haben

Bevor er *A* ausführt, muss der Agent:

1. die **Wahrscheinlichkeiten** $P(Result_i(A)|Do(A),E)$ bestimmen
2. die **expected utility** von *A*, gegeben Evidenz *E* berechnen:

   $EU(A|E)=\Sigma_i P(Result_i(A)|Do(A),E)\ U(Result_i(A))$

   mit $U(S)$ **utility function** von Zustand S
3. die Aktion *A* auswählen, welche *EU* maximiert

Decision-Making

Lösung eines Entscheidungsproblems

=

Bestimmen der optimal erreichbaren *expected utility*, zusammen mit entsprechenden Handlungssequenzen

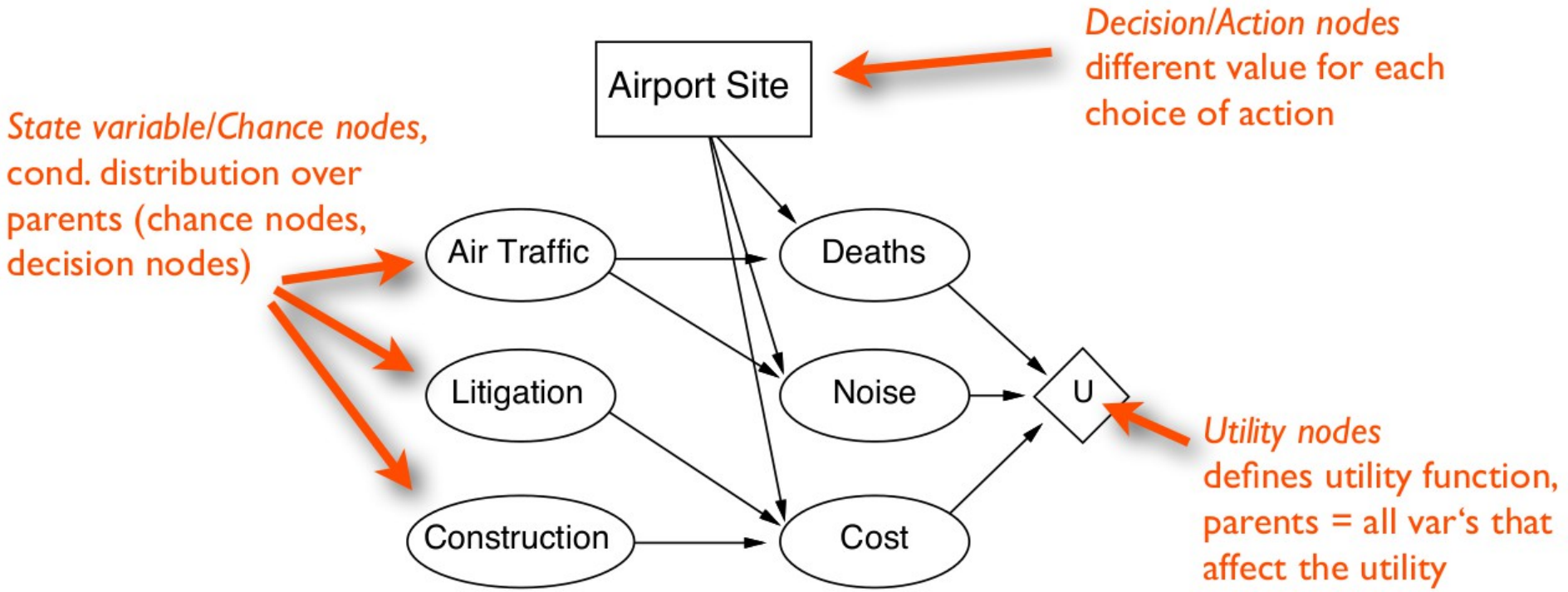
MEU-basiertes Decision-Making kann auf verschiedene Weisen modelliert werden:

- Decision Trees (letzte Woche)
- Bayesian Decision Networks
- Markov Decision Problem

Bayesian Decision Networks (BDN)

Erweitern Bayes-Netze um **Aktionen** und **Utilities**

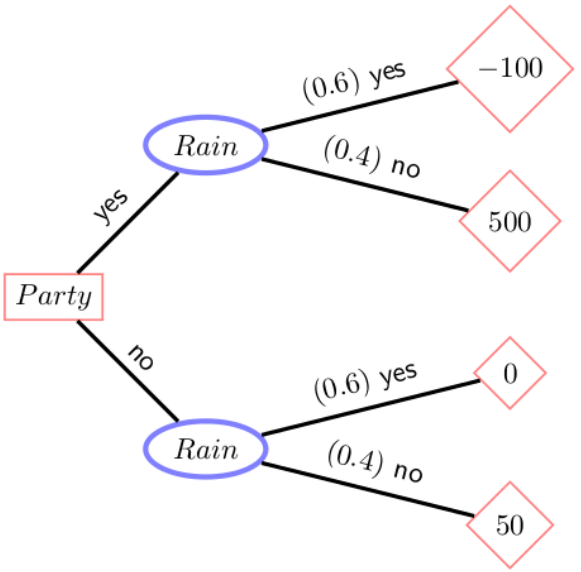
(werden auch *influence diagram* (ID) genannt)



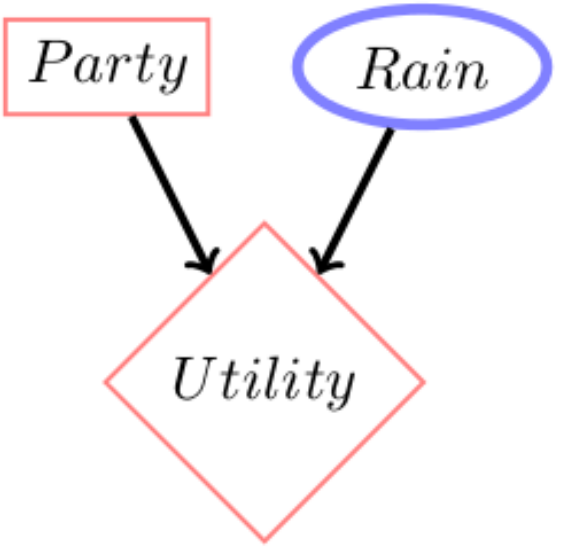
Bayesian Decision Networks (BDN)

Party-Beispiel

Decision Tree



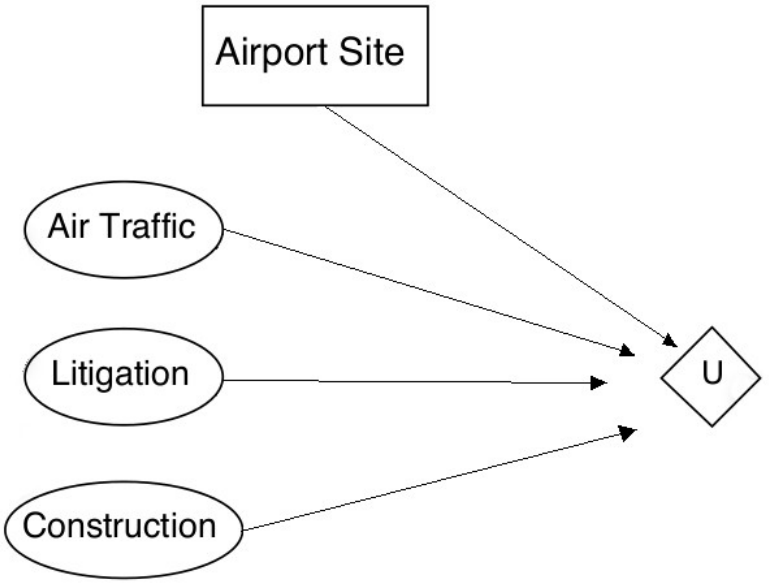
Decision Network



Evaluation von Decision Networks

Eigenschaften

- Future States (nach Decision Node, im Beispiel: *Cost, Noise, Deaths*) erhalten Werte nicht über Evidenz
- *Cost*, *Noise* und *Deaths* können für eine explizite Beschreibung ausgelassen werden → unflexiblere Utilty Function (stattdessen direkter Einfluss von Decision- und Chance-Nodes auf Utility-Node)



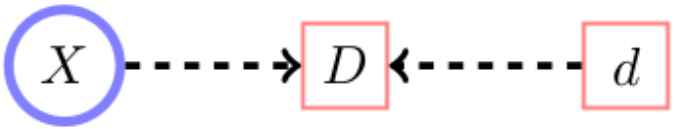
Evaluation von Decision Networks

Algorithmus

1) Zuerst: Alle **Evidenz**-Variablen für den aktuellen Zustand setzen
2) Dann: **für jeden möglichen Wert** des Decision Nodes:
    I. Setze **Decision Node** auf diesen Wert
    II. Berechne die **A-posteriori**-Wahrscheinlichkeit **aller Parent Nodes des Utility Nodes** (durch Bayes-Netz-Inferenz)
    III. Berechne die (**Expected**) **Utility** für die Aktion
3) Wähle die **Aktion,** welche die **Expected Utilty maximiert**

Partial Ordering

Information Link: von Chance Nodes *X* und anderen Decision Nodes *d* zum Decision Node *D*

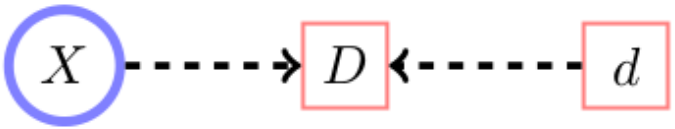


Der Wert von *X* ist bekannt, bevor *D* entschieden wird

D steht nicht in *funktionaler* Verbindung mit Elter *X*

Partial Ordering

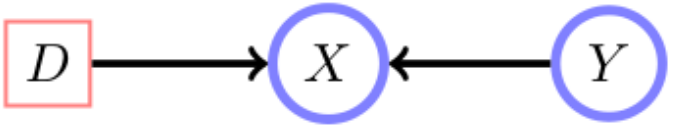
Information Link: von Chance Nodes *X* und anderen Decision Nodes *d* zum Decision Node *D*



Der Wert von *X* ist bekannt, bevor *D* entschieden wird

D steht nicht in *funktionaler* Verbindung mit Elter *X*

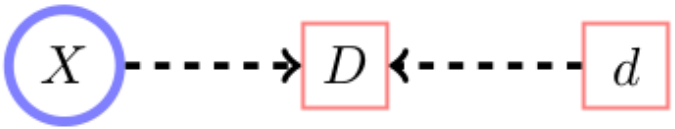
Chance Nodes: Können von Decision Nodes und anderen Chance Nodes abhängig sein



Werden sichtbar, wenn *D* entschieden wurde

Partial Ordering

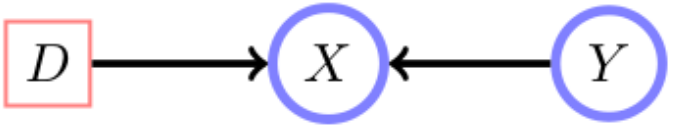
Information Link: von Chance Nodes *X* und anderen Decision Nodes *d* zum Decision Node *D*



Der Wert von *X* ist bekannt, bevor *D* entschieden wird

D steht nicht in *funktionaler* Verbindung mit Elter *X*

Chance Nodes: Können von Decision Nodes und anderen Chance Nodes abhängig sein



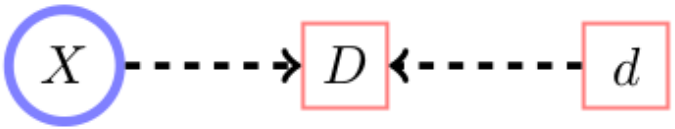
Werden sichtbar, wenn *D* entschieden wurde

Utilities: Deterministische Funktion aus Elternknoten

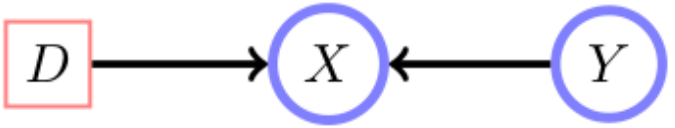
Eltern können sowohl Chance Nodes als auch Decision Nodes sein

Partial Ordering

Information Link: von Chance Nodes *X* und anderen Decision Nodes *d* zum Decision Node *D*



Chance Nodes: Können von Decision Nodes und anderen Chance Nodes abhängig sein



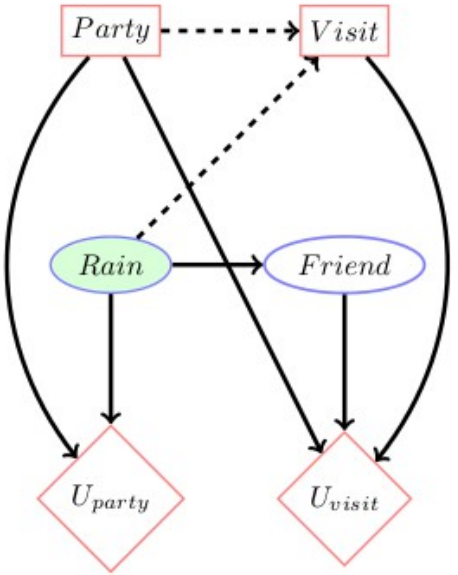
Utilities: Deterministische Funktion aus Elternknoten

Partial Ordering bekannter Variablen X_i und Entscheidungen D_i:

$$\mathcal{X}_0 \prec D_1 \prec \mathcal{X}_1 \prec D_2, \ldots, \prec \mathcal{X}_{n-1} \prec D_n \prec \mathcal{X}_n$$

Causal Consistency

- Aktuelle Entscheidungen (Decision Nodes) können die Vergangenheit (Chance Nodes) nicht beeinfluss
- Folglich steht jede Zufallsvariable (Chance Node) im Partial Ordering nach der Entscheidung D, von der sie abhängig ist
- Unter der Annahme des nicht-Vergessens bedeutet dies für valide Decision Networks, dass ein gerichteter Pfad alle Entscheidungen verbinden muss.

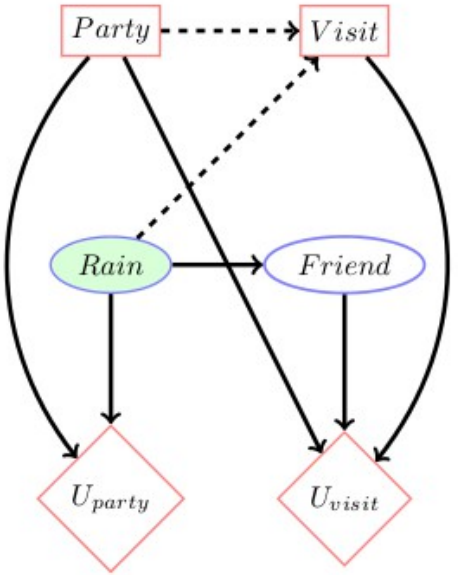


$$Party^* \prec Rain \prec Visit^* \prec Friend$$

Decision Networks lösen

Die optimale **Expected Utility** erhält man durch **Summierung** über nicht einsehbare (*unrevealed*) Zufallsvariablen (*Chance Nodes*) und durch **Optimierung** über zukünftige Entscheidungen (*Decision Nodes*).

$$\max_{Party} \sum_{Rain} p(Rain) \max_{Visit} \sum_{Friend} p(Friend|Rain)$$
$$\times \left[U_{party}(Party, Rain) + U_{visit}(Visit, Friend)\mathbb{I}\left[Party = \text{no}\right]\right]$$



$$Party^* \prec Rain \prec Visit^* \prec Friend$$

Should I do a PhD?

Consider a decision whether or not to do PhD as part of our education (E). Taking a PhD incurs costs, U_C both in terms of fees, but also in terms of lost income. However, if we have a PhD, we are more likely to win a Nobel Prize (P), which would certainly be likely to boost our Income (I), subsequently benefitting our finances (U_B). The ordering is (excluding empty sets)

$$E^* \prec \{I, P\}$$

$\mathrm{dom}(E) = (\text{do PhD}, \text{no PhD})$, $\mathrm{dom}(I) = (\text{low}, \text{average}, \text{high})$, $\mathrm{dom}(P) = (\text{prize}, \text{no prize})$.

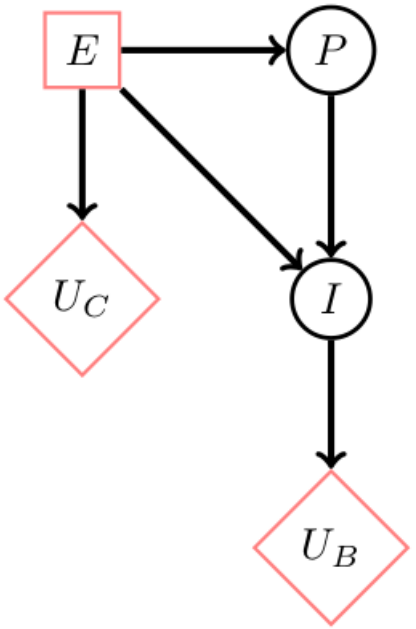
$$p(\text{win Nobel prize}|\text{no PhD}) = 0.0000001 \qquad p(\text{win Nobel prize}|\text{do PhD}) = 0.001$$

$p(\text{low}|\text{do PhD}, \text{no prize}) = 0.1$ $p(\text{average}|\text{do PhD}, \text{no prize}) = 0.5$ $p(\text{high}|\text{do PhD}, \text{no prize}) = 0.4$
$p(\text{low}|\text{no PhD}, \text{no prize}) = 0.2$ $p(\text{average}|\text{no PhD}, \text{no prize}) = 0.6$ $p(\text{high}|\text{no PhD}, \text{no prize}) = 0.2$
$p(\text{low}|\text{do PhD}, \text{prize}) = 0.01$ $p(\text{average}|\text{do PhD}, \text{prize}) = 0.04$ $p(\text{high}|\text{do PhD}, \text{prize}) = 0.95$
$p(\text{low}|\text{no PhD}, \text{prize}) = 0.01$ $p(\text{average}|\text{no PhD}, \text{prize}) = 0.04$ $p(\text{high}|\text{no PhD}, \text{prize}) = 0.95$

The utilities are

$$U_C(\text{do PhD}) = -50000, \quad U_C(\text{no PhD}) = 0,$$
$$U_B(\text{low}) = 100000, \quad U_B(\text{average}) = 200000, \quad U_B(\text{high}) = 500000$$

Should I do a PhD?



The expected utility of Education is

$$U(E) = \sum_{I,P} p(I|E,P)p(P|E)\left[U_C(E) + U_B(I)\right]$$

so that $U(\text{do phd}) = 260174.000$, whilst not taking a PhD is $U(\text{no phd}) = 240000.0244$, making it on average beneficial to do a PhD.

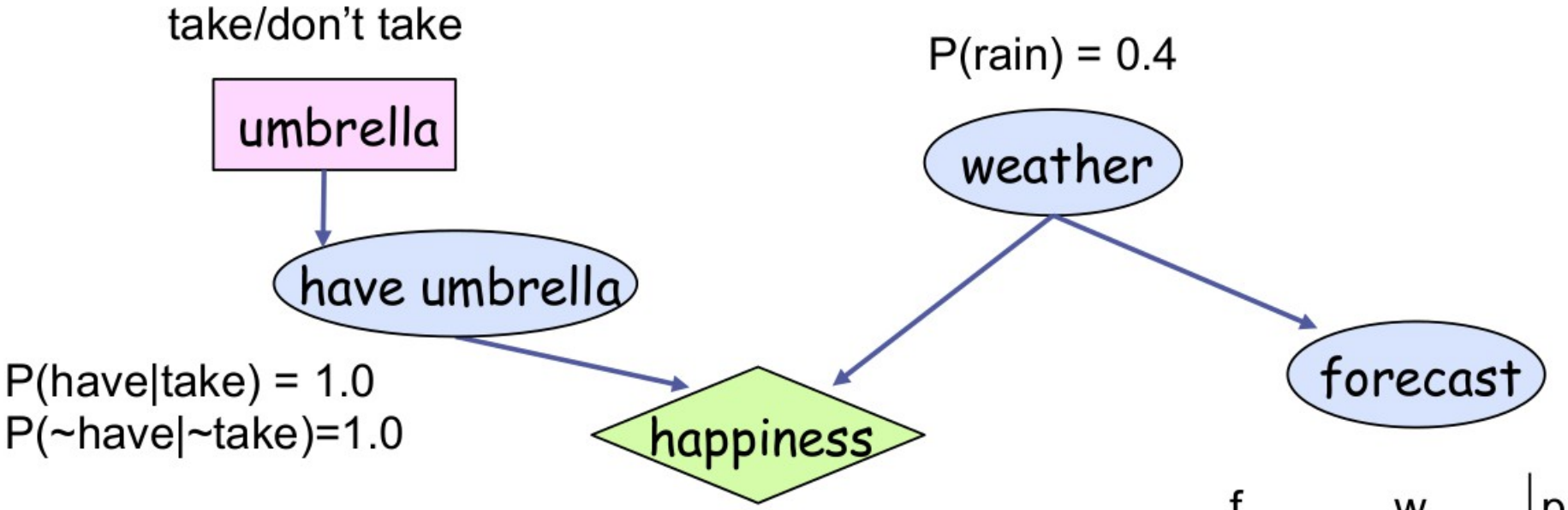
Value of Information

- Bisherige Annahme: Relevante Informationen sind zum Zeitpunkt der Entscheidung vorhanden
- Praktisch seltener der Fall
- Beispiel: Ein Patient kommt zum Arzt. Er kann den Patienten nicht jedem denkbaren Test unterziehen und Antworten auf alle möglichen Fragen bekommen. Stattdessen stellt der Arzt Fragen und führt Tests durch, die bzgl. Zeit, Risiko und Kosten von Wert sind

Decision-Making: Umbrella Network

Queries: Should I take my umbrella?

What is the value of knowing the weather forecast?



f	w	p(f|w)
sunny	rain	0.3
rainy	rain	0.7
sunny	no rain	0.8
rainy	no rain	0.2

Value of Information - Berechnung

Gegeben: Aktuelle Evidenz *E*, aktuell beste Aktion *α*

Annahmen: Ergebnisse S_i für *α*, mögliche neue Evidenz E_j

$$EU(\alpha|E) = \max_a \sum_i U(S_i)\ P(S_i|E, a)$$

Value of Information - Berechnung

Gegeben: Aktuelle Evidenz *E*, aktuell beste Aktion *α*

Annahmen: Ergebnisse S_i für *α*, mögliche neue Evidenz E_j

$$EU(\alpha|E) = \max_a \sum_i U(S_i)\ P(S_i|E,a)$$

Angenommen, $E_j = e_{jk}$ *ist bekannt, dann wähle* α_{ejk}*, s.d.*

$$EU(\alpha_{e_{jk}}|E, E_j = e_{jk}) = \max_a \sum_i U(S_i)\ P(S_i|E,a,E_j = e_{jk})$$

Value of Information - Berechnung

Gegeben: Aktuelle Evidenz *E*, aktuell beste Aktion *α*

Annahmen: Ergebnisse S_i für *α*, mögliche neue Evidenz E_j

$$EU(\alpha|E) = \max_a \sum_i U(S_i)\ P(S_i|E,a)$$

Angenommen, $E_j = e_{jk}$ *ist bekannt, dann wähle* α_{ejk}*, s.d.*

$$EU(\alpha_{e_{jk}}|E, E_j = e_{jk}) = \max_a \sum_i U(S_i)\ P(S_i|E, a, E_j = e_{jk})$$

E_j ist eine Zufallsvariable ist, deren Wert aktuell nicht bekannt ist. Um den Wert für E_j zu errechnen, gegeben Evidenz E, wird über alle möglichen Werte von e_{jk} gemittelt. Das Resultat ist der **Value of Perfect Information**:

$$VPI_E(E_j) = \left(\sum_k P(E_j = e_{jk}|E) EU(\alpha_{e_{jk}}|E, E_j = e_{jk})\right) - EU(\alpha|E)$$

Information-Gathering Agent

Der Agent wählt wiederholt die Evidenz mit der größten möglichen Information aus. Er macht dies so lange, bis die Kosten für die nächste Evidenz größer ist als der erwartete Nutzen:

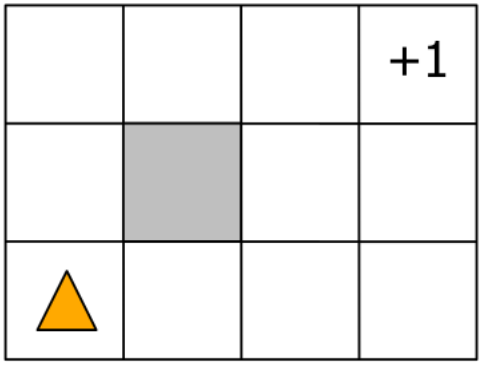
```
function INFORMATION-GATHERING-AGENT(percept) returns eine Aktion
   static: D, ein Entscheidungsnetzwerk

   integriere percept in D
   j ← der Wert, der WPI(E_j) − Kosten(E_j) maximiert
   if WPI(E_j) > Kosten(E_j)
      then return REQUEST(E_j)
   else return die beste Aktion aus D
```

Komplexe Entscheidungsprobleme

- Utilities, Uncertainty und Sensing
- Utility hängt von einer **Sequenz von Aktionen** ab
- Problemstellung generalisiert Such- und Planprobleme zum Prozess des Findens einer passenden **Action Policy**

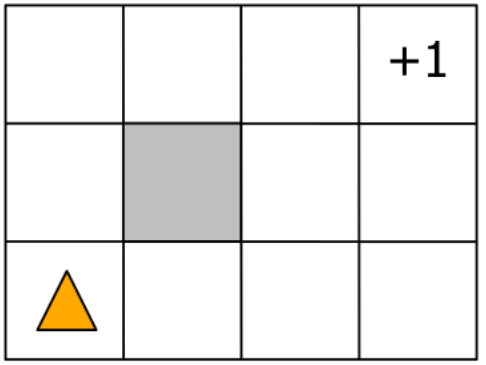
Einfaches Navigationsproblem



In jedem Zustand gibt es die möglichen **Aktionen** *U (hoch), D (runter), R (rechts)* und *L (links)*.

Der **Zielzustand** ist mit *+1* markiert.

Einfaches Navigationsproblem



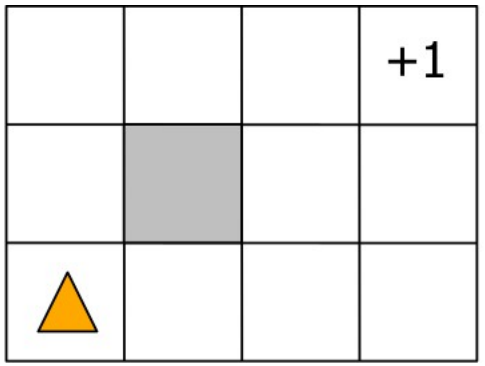
In jedem Zustand gibt es die Möglichen **Aktionen** *U (hoch), D (runter), R (rechts)* und *L (links)*.

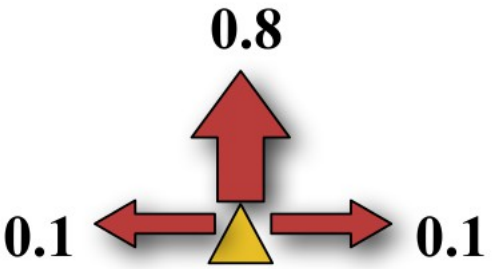
Der **Zielzustand** ist mit *+1* markiert.

Die Umgebung ist **voll observabel** *(fully observable)* und **deterministisch**:

Lösung = Sequenz der Aktionen (U,U,R,R,R)

Probabilistisches Übergangsmodell

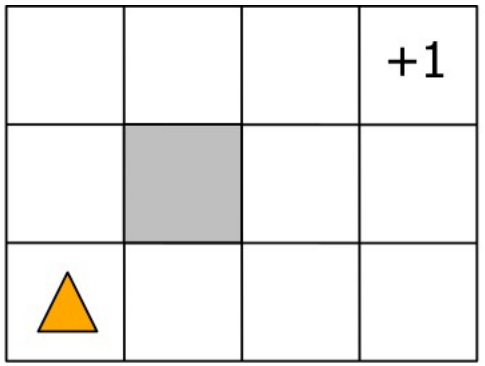


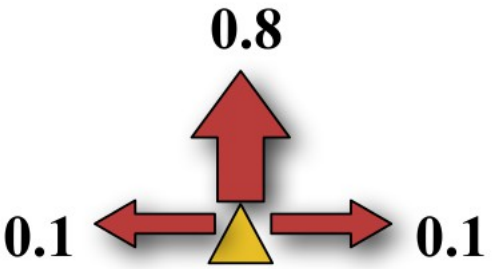


Nicht-deterministische Aktionen durch stochastische Bewegung:

- Die geplante Bewegung wird zu einer Wahrscheinlichkeit von 0,8 ausgeführt. In den restlichen Fällen weicht der Agent zu jeweils 0,1 nach rechts/links, relativ zur Bewegungsrichtung, ab.
- Wenn der Agent gegen eine Wand läuft, verbleibt er in seiner aktuellen Position.

Probabilistisches Übergangsmodell

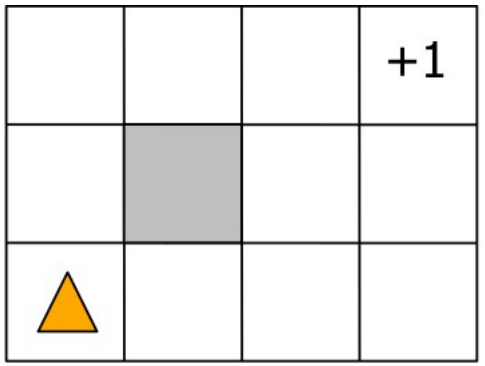


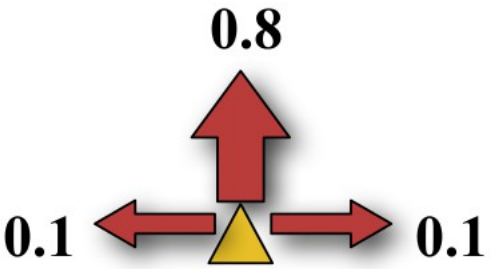


Nicht-deterministische Aktionen durch stochastische Bewegung:

- Die Sequenz (U,U,R,R,R) erreicht ihr Ziel mit einer Wahrscheinlichkeit von $0,8^5 = 0,32768$
- Oder versehentlich mit $0,1^4 \times 0,8 = 0,00008$
- Gesamtwahrscheinlichkeit, das Ziel zu erreichen = 0,32776

Probabilistisches Übergangsmodell





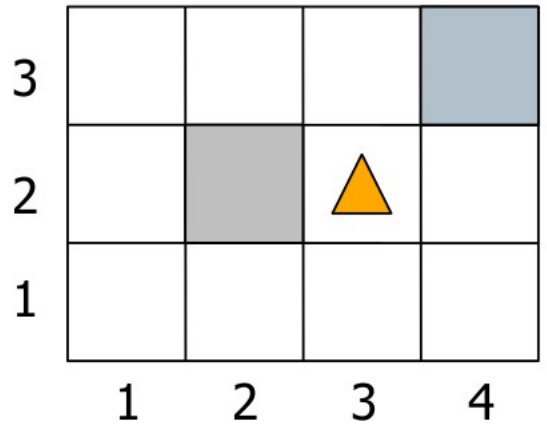
Übergangsmodell *(Transition Model)*:

T(s,a,s') = Wahrscheinlichkeit mit Aktion *a* von *s* nach *s′* zu kommen

Markov-Eigenschaft:

Überganswahrscheinlickeiten hängen nur vom aktuellen Zustand ab, aber nicht davon, wie dieser Zustand erreicht wurde. (Gedächtnislosigkeit)

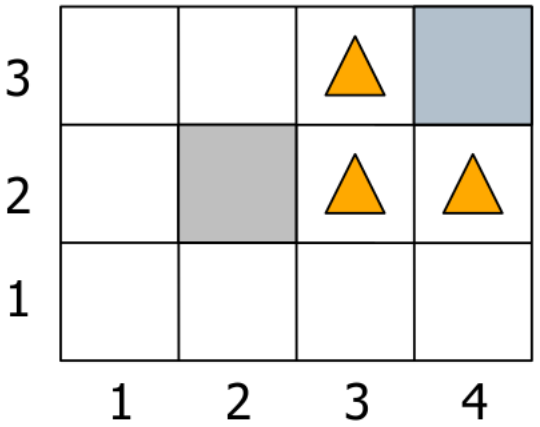
Aktionssequenz

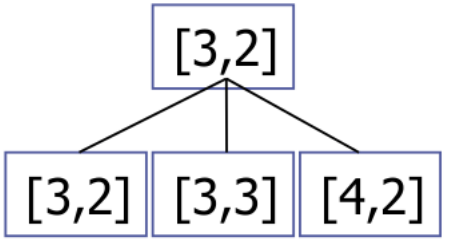


Position: [3,2]

Geplante Aktionssequenz: (U,R)

Aktionssequenz

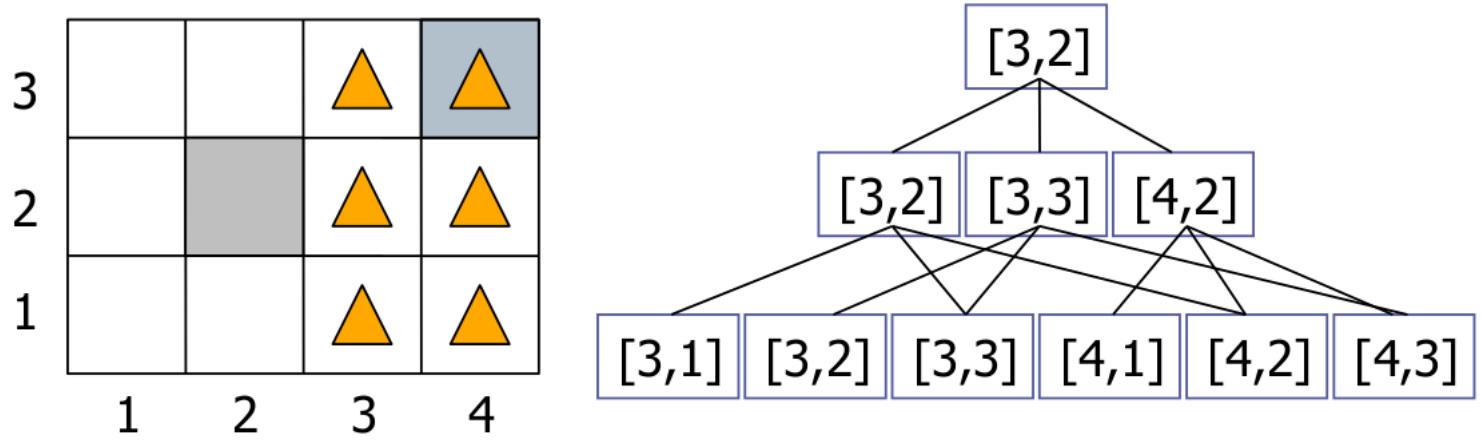




Geplante Aktionssequenz: (U,R)

U ist ausgeführt

Aktionssequenz

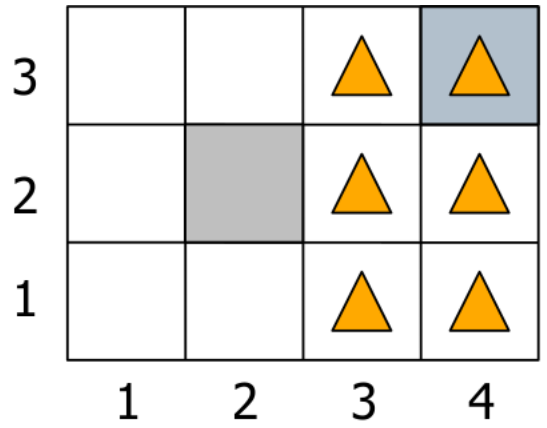


Geplante Aktionssequenz: (U,R)

U, R ist ausgeführt

Es gibt nun 9 mögliche Zustands-Sequenzen – sog. **Histories** – und 6 mögliche Endzustände des Agenten

Aktionssequenz



P([4,3] | (U,R).[3,2]) =
P([4,3] | R.[3,3])
x **P**([3,3] | U.[3,2])
+ **P**([4,3] | R.[4,2])
x **P**([4,2] | U.[3,2])

P([3,3] | U.[3,2]) = 0.8
P([4,2] | U.[3,2]) = 0.1

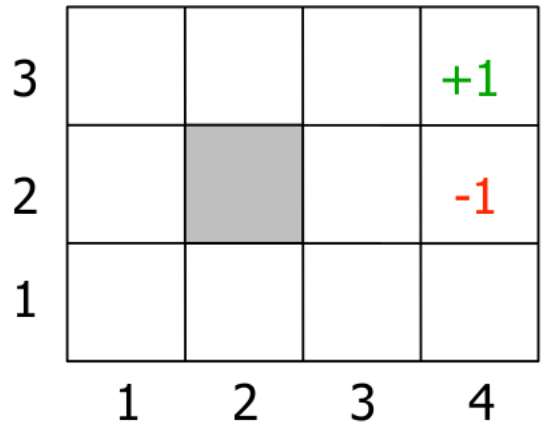
P([4,3] | R.[3,3]) = 0.8
P([4,3] | R.[4,2]) = 0.1

P([4,3] | (U,R).[3,2]) = 0.65

Anmerkung:

In der Gleichung P([4,3] | (U,R).[3,2]) sieht man die Wichtigkeit der Markov-Eigenschaft.

Utility-Funktion



Der Agent erhält in jedem Zustand *S* einen Reward R(S)

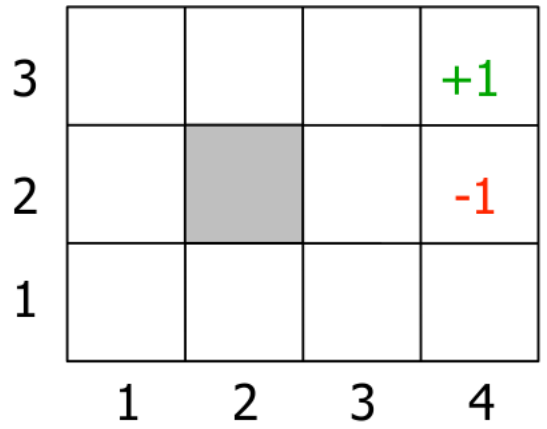
[4,3] und [4,2] sind **Zielzustände**

[4,3] enthält einen **Power Supply** mit Reward +1

[4,2] ist eine **Sandgrube** mit Reward -1

Jedes andere Feld hat einen Reward von -0,04

Utility-Funktion



Der Agent erhält in jedem Zustand *S* einen Reward R(S)

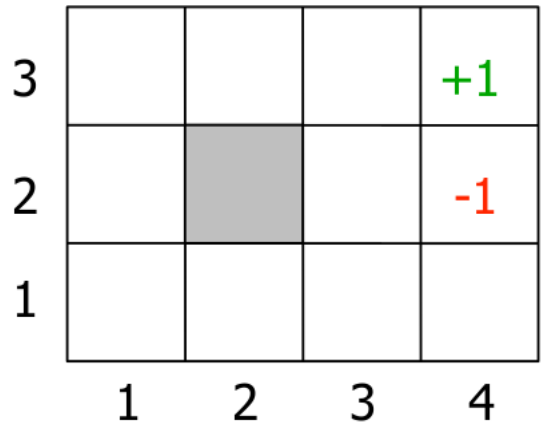
[4,3] und [4,2] sind **Zielzustände**

[4,3] enthält einen **Power Supply** mit Reward +1

[4,2] ist eine **Sandgrube** mit Reward -1

Jedes andere Feld hat einen Reward von -0,04

Utility der History

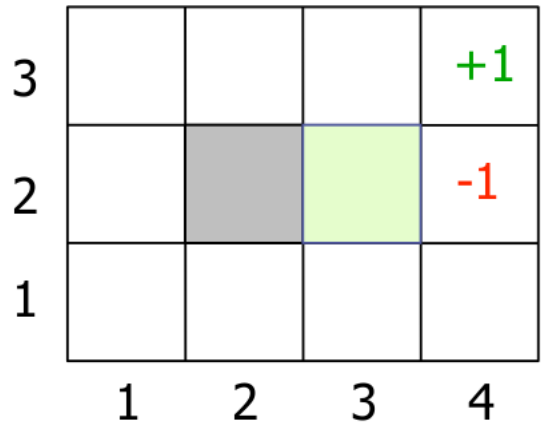


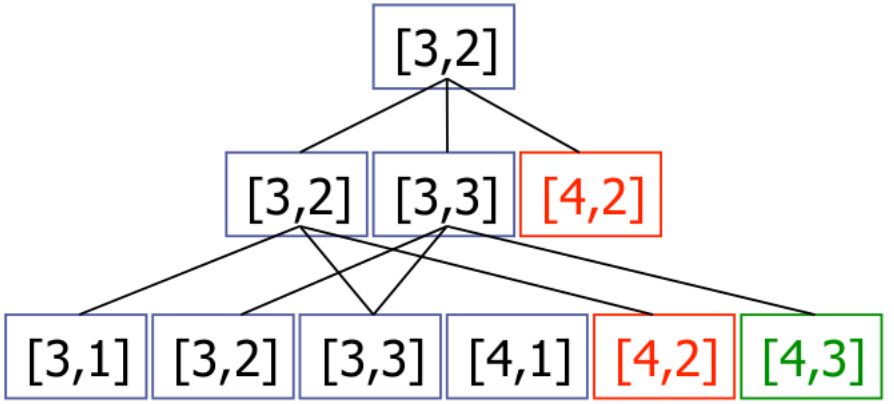
Die **Utility** wird durch die **History** bestimmt.

Utility = Utility der Zielzustände (+1 oder -1) *PLUS* Summe der Rewards besuchter Felder (**additiver Reward**)

Der Agent will die Umgebung so schnell wie möglich verlassen!

Utility der History



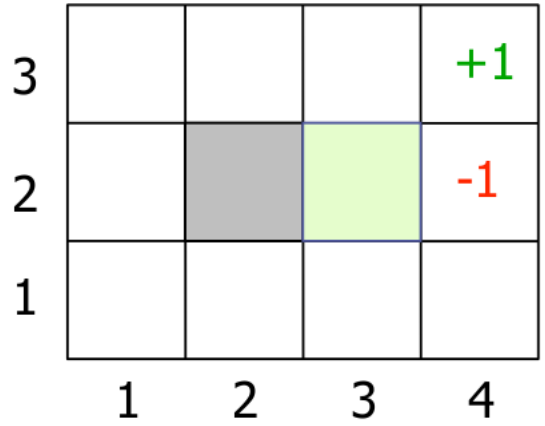


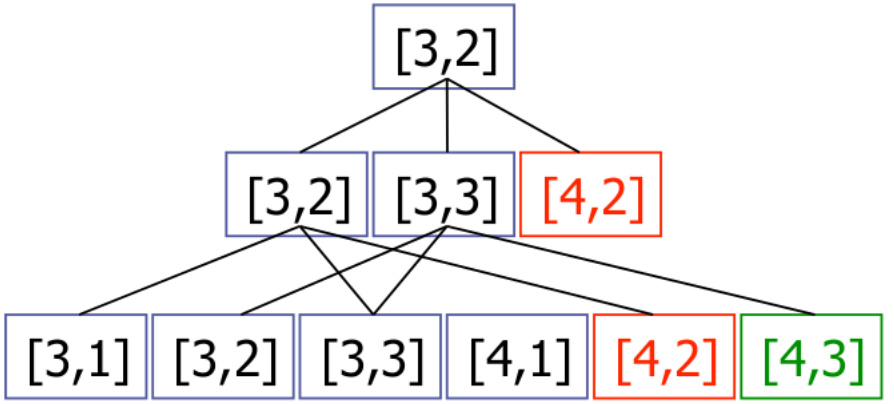
Die Sequenz (U,R), von [3,2] ausgehend, hat 7 mögliche **Histories** (Aktionssequenzen).

Jede davon hat eine **Wahrscheinlichkeit** P_h und eine **additive Utility der History**:

$$U_h = R(s_0)+R(s_1)+\ldots+R(s_n) = \sum R(s_i)$$

Utility einer Aktionssequenz





Utility einer History:

$$\mathbf{U}_h = R(s_0)+R(s_1)+\ldots+R(s_n) = \sum R(s_i)$$

Utility einer Aktionssequenz (= Exptected Utility der Histories):

$$\mathbf{U} = \Sigma_h \mathbf{U}_h \mathbf{P}(h)$$

Optimale Aktionssequenz

Die Sequenz (U,R), von [3,2] ausgehend, hat 7 mögliche **Histories** (Aktionssequenzen).

Die Utility einer Sequenz ist die Expected Utility der Histories.

Die **optimale Sequenz** ist diejenige Sequenz mit der maximalen Utility.

Will man immer die optimale Sequenz berechnen?

Optimale Aktionssequenz

Die Sequenz (U,R), von [3,2] ausgehend, hat 7 mögliche **Histories** (Aktionssequenzen).

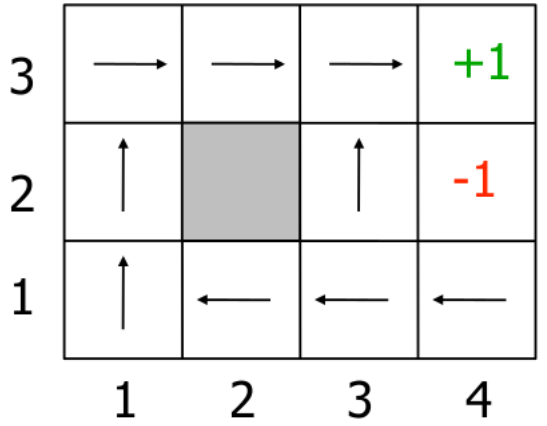
Die Utility einer Sequenz ist die Expected Utility der Histories.

Die **optimale Sequenz** ist diejenige Sequenz mit der maximalen Utility.

Will man immer die optimale Sequenz berechnen?

- Nur wenn die Sequenz **blind** ausgeführt werden soll.
- Eine Lösung muss **für jeden Zustand**, den ein Agent erreichen kann, **eine Aktion** spezifizieren.

Policy



- Eine **Policy Π** ist eine Abbildung von Zuständen auf Aktionen
- Eine Policy ist **vollständig**, wenn sie für jeden möglichen Zustand der Welt definiert ist

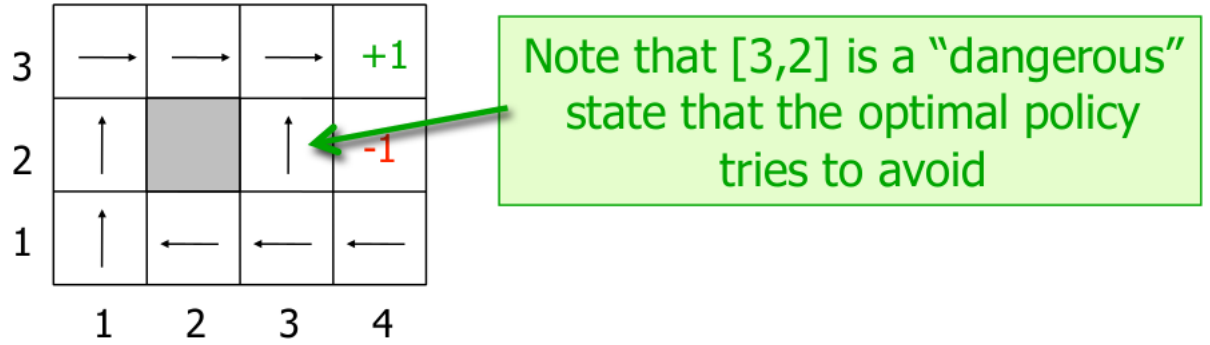
Reaktiver Agent

Repeat:

1. s ← sensed state
2. If s is terminal state then exit
3. a ← Π(s)
4. Perform a

R&N: „policy is a description of a simple reflex agent, computed from information used for a utility-based agent“

Optimale Policy



- Eine **optimale Policy Π*** liefert immer eine **History** mit **maximaler Expected Utility.**
- Eine optimale Policy **wägt** in Abhängigkeit der Werte R(S) von nicht-Zielzuständen **Risiken** und **Rewards** ab.

Wie berechnet man die optimale Policy Π*?

Markov Decision Problem (MDP)

Spezifikation für **sequenzielle Entscheidungsprobleme in voll observablen** *(fully observable)* Umgebungen:

1. **Startzustand** s_0
2. **Übergangsmodell** $T(s,a,s') = \boldsymbol{P}(s'|a,s)$
3. **Reward-Funktion**: $R(s)$ (oder $R(s,a,s')$, **additiv**)

Wie berechnet man die optimale Policy Π*?

Dieses Problem ist ein

***Markov Decision Problem* (MDP)**