

---

# Einführung in die Künstliche Intelligenz

## WS 10/11 - Prof. Dr. J. Fürnkranz



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

---

### Beispiellösung für das 1. Übungsblatt (03.11.2010)

---

#### Aufgabe 1 Agenten-Umgebung

---

- a) Eine beispielhafte PEAS-Beschreibung für eine Internet Suchmaschine:
- Performance:
    - Aus Sicht des Users: Relevanz der Suchergebnisse, Precision, Recall, Ranking Qualität, Beantwortungszeit, Vollständigkeit der Ergebnisse, Usability
    - Aus Sicht des Betreibers: Crawlinggeschwindigkeit, speichereffiziente Indizierung
  - Environment: World-Wide-Web, Hyperlinks, verschiedene Medien (Text, Bilder, Videos), verschiedene Sprachen, verschiedene HTML Versionen, inkorrekte HTML Codes (Fehlertoleranz), mehrdeutige Eingaben/Wörter, Schreibfehler, Spam
  - Actuators: Weiterleitung auf relevante Seiten, indirekte Beeinflussung: Pagerank, Suchmaschinenoptimierung, ...
  - Sensors: Text-Eingabe, Click-Messung, Parsen von verschiedenen Medien im WWW
- b) Eine mögliche Charakterisierung der obigen Umgebung ist:
- Fully observable vs. partially observable: partially observable, eine nicht gelinkte Seite wird nicht indiziert. Neben der Problematik die Abdeckung aller Websites zu maximieren, besteht das grundsätzliche Problem, dass der semantische Inhalt einer Website/Suchanfrage nicht erfasst wird.
  - Deterministic vs. stochastic: stochastic.
  - Episodic vs. sequential: sequential.
  - Static vs. dynamic: dynamic.
  - Discrete vs. Continuous: continuous.
  - Single agent vs. multiagent: multiagent.

---

#### Aufgabe 2 Agenten-Struktur

---

Machen Sie sich klar, worin sich die Agenten-Typen *reflex*, *model-based*, *goal-based* und *utility-based* unterscheiden:

- Im folgenden werden nur reflex vs. model-based, model-based vs. goal-based und goal-based vs. utility-based Agenten verglichen.
  - reflex vs. model-based: der model-based Agent unterhält im Gegensatz zum reflex Agent ein Modell von der Umgebung. Dies ist hilfreich, z.B. wenn die Umgebung nur teilweise erfassbar ist, da mehrfache Sensormessungen (zeitlich verschieden oder verschiedene Sensorinformationen) eine potentiell bessere Abbildung der Umgebung ermöglichen.
  - model-based vs. goal-based: Es werden beim goal-based Agenten Ziele statt condition-action rules definiert. Es ist nun vielmehr Aufgabe des Agenten geeignete Regeln zu bestimmen oder allgemeiner Aktionen durchzuführen, die zum vorgegebenen Ziel führen.
  - goal-based vs. utility-based: Durch die Einführung einer Bewertungsfunktion wird es dem Agenten ermöglicht Zustände zu bewerten. Im allgemeinen wird der Agent seine Aktionen so wählen, so dass er seine Bewertungsfunktion maximiert. Stimmt die Bewertungsfunktion mit dem Performance Kriterium überein, agiert es rational.

---

---

### Aufgabe 3 Modellierung

---

#### a) Ketten und Zustände

Kette:  $(f, l, s)$   $f$  Form (*linie/kreis*),  
 $l$  Länge,  
 $s$  Status (*auf/zu*)

Zustand:  $\{(n, f, l, s), (n', f', l', s'), \dots\}$   $n$  Anzahl vorhandener  $(f, l, s)$ -Ketten

#### Operatoren

AUFMACHEN( $f, l$ ): verwandelt eine Kette  $(f, l, zu)$  in zwei Ketten  $(linie, l - 1, zu)$  und  $(linie, 1, auf)$

VERBINDEN( $l', f$ ): verwandelt drei Ketten  $(linie, l, zu)$ ,  $(linie, l', zu)$  und  $(linie, 1, auf)$  in eine Kette  $(f, l + l' + 1, zu)$

Merke:  $(f, l, s)$  mit  $l = 0$  ist eine leere Kette!

#### b)

Ausgangszustand:  $\{(4, linie, 3, zu)\}$

Zielzustand:  $\{(1, kreis, 12, zu)\}$

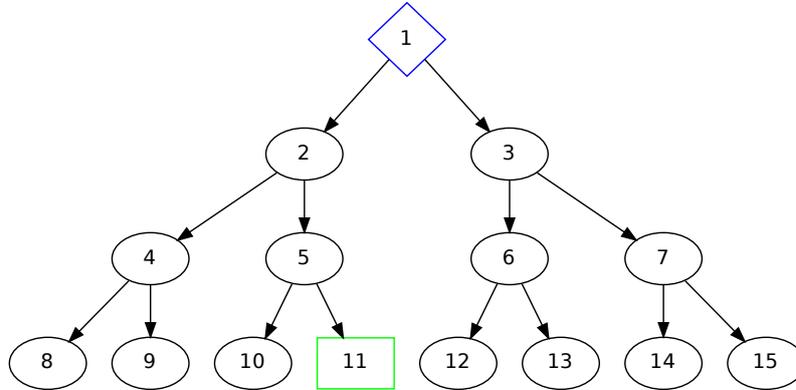
#### c) Lösungspfad

$\{(4, linie, 3, zu)\}$	
↓	
$\{(3, linie, 3, zu), (1, linie, 2, zu), (1, linie, 1, auf)\}$	AUFMACHEN( <i>linie, 3</i> )
↓	
$\{(3, linie, 3, zu), (1, linie, 1, zu), (2, linie, 1, auf)\}$	AUFMACHEN( <i>linie, 2</i> )
↓	
$\{(3, linie, 3, zu), (3, linie, 1, auf)\}$	AUFMACHEN( <i>linie, 1</i> )
↓	
$\{(1, linie, 7, zu), (1, linie, 3, zu), (2, linie, 1, auf)\}$	VERBINDEN( <i>3, 3, linie</i> )
↓	
$\{(1, linie, 11, zu), (1, linie, 1, auf)\}$	VERBINDEN( <i>7, 3, linie</i> )
↓	
$\{(1, kreis, 12, zu)\}$	VERBINDEN( <i>11, 0, kreis</i> )

### Aufgabe 4 Uninformierte Suchalgorithmen

Betrachten Sie einen Zustandsraum, wo der Ausgangszustand die Nummer 1 ist und die Nachfolgerfunktion für den Zustand mit der Nummer  $n$  zwei Zustände mit den Nummern  $2n$  und  $2n + 1$  zurückgibt.

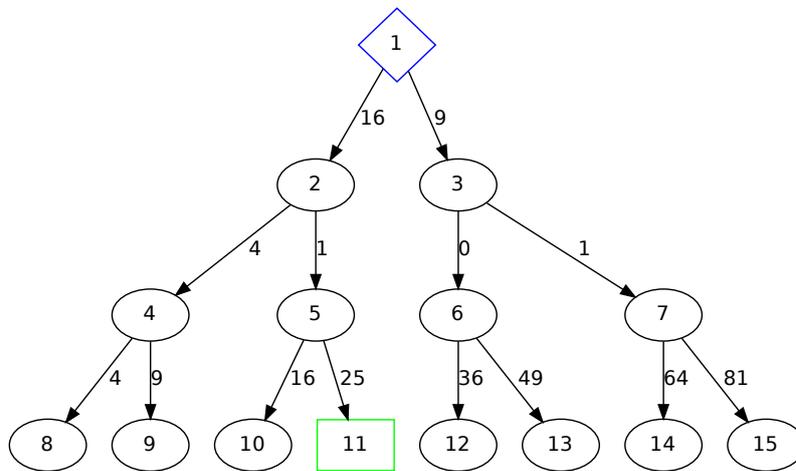
a) Zeichnen Sie den Teil des Zustandsraumes für die Zustände 1 bis 15.



b) Der Zielzustand sei 11. Geben Sie die Reihenfolge an, in der die Knoten mit den folgenden Suchalgorithmen besucht werden:

1. Breadth-First: 1,2,3,4,5,6,7,8,9,10,11
2. Depth-First: 1,2,4,8,9,5,10,11
3. Iterative-Deepening: 1, 1,2,3, 1,2,4,5,3,6,7, 1,2,4,8,9,5,10,11

c) Die Kosten von einem unmittelbaren Vorgänger-Knoten zum Knoten mit der Nummer  $n$  zu gelangen sei  $(n - 6)^2$ . Geben Sie die Reihenfolge an, in der die Knoten mit der *Uniform-Cost-Search* besucht werden.



1(0),3(9),6(9), 7(10), 2(16), 5(17), 4(20), 8(24), 9(29), 10(33), 11(42)

Die restlichen Knoten haben folgende Werte: 12(45), 13(58), 14(74), 15(91). Die Werte in den Klammern beschreiben die Pfadkosten.