

# Praktikum Anwendungen wissensbasierter Systeme - Intelligente Systeme

## Wumpus-Welt

**Abgabetermin:** Freitag, 03.06.2005

Auf diesem Blatt soll ein modellbasierter Agent für die Wumpus-Welt geschrieben werden.

In der Datei *wumpus.tar.gz* findet ihr alle Vorgaben zu dieser Aufgabe. Im einzelnen sind das die Verzeichnisse

- *aima*, das allgemeine Funktionen für die Wumpus-Welt enthält und
- *paip*, das die Dateien für Prolog enthält

und die Dateien

- *environment.lisp* definiert die Umgebung,
- *knowledge-base.lisp* enthält die Funktionen `tell`, `ask`, `bindings-of` und `reset-kb`,
- *wumpus-agent-aa.fasl* enthält unseren Wumpus-Agenten zum Testen.

An diesen Verzeichnissen und Dateien müsst ihr nichts ändern oder sie aufrufen, sie sind nur nötig, damit das System läuft. Für euch sind vor allem diese Dateien interessant:

- *start.lisp* zum automatischen Laden aller benötigten Dateien
- *wumpus-agent.lisp* enthält die Definition eurer Agentenfunktion.

In der Datei *wumpus-agent.lisp* wird ein Wumpus-Agent definiert. Da wir am Ende alle Wumpus-Jäger gegeneinander laufen lassen wollen, sollte jede Gruppe ihrem Agenten einen eigenen Namen geben.

Um eure Agentenfunktion zu testen, gibt es die Funktion `run`. Diese erhält als Parameter den Namen eures Agenten. Außerdem können folgende key-Parameter angegeben werden:

**:p-pit** Häufigkeit der Gruben. Der Standardwert ist 0.2.

**:world** Ohne Angabe dieses Parameters wird der Agent in eine zufällig erzeugte Welt gesetzt. Oft kann es aber nützlich sein, wenn man mehrere Male die gleiche Welt ausprobieren kann. Die möglichen Welten sind durchnummeriert und hier kann die Nummer der gewünschten Welt übergeben werden.

Um eine Welt auszuwählen, kann man sich mit der Funktion `show-worlds` alle Welten mit ihren dazugehörigen Nummern ausgeben lassen. Diese Funktion erwartet als Parameter eine Zahl, die die Nummer der größten angezeigten Welt darstellt. Außerdem sind folgende key-Parameter möglich:

**:from** gibt die Nummer der ersten Welt an, die ausgegeben werden soll.

**:p-pit** gibt die Häufigkeit der Gruben an (siehe auch `run`).

```
(show-worlds 20 :from 5 :p-pit 0.15)
```

gibt alle Welten von 5 bis 20 mit Grubenwahrscheinlichkeit 0.15 aus.

**:max-steps** begrenzt die Anzahl der Schritte, die ausgeführt werden sollen. Der Standardwert ist 1000.

**:stream** Dieser Parameter kann auf `nil` gesetzt werden, wenn man die Ausgabe der einzelnen Zeitschritte verhindern und nur das Endergebnis sehen will.

```
(run 'wumpus-agent-aa  
:p-pit 0.15  
:world 10  
:max-steps 20)
```

Um einen Agenten mit anderen zu vergleichen, gibt es die Funktion `agent-trials`, die verschiedene Umgebungen generiert und alle Agenten darin laufen lässt und jeweils die Punkte der Bewertungsfunktion aufsummiert. Am Ende werden wir einen Wettbewerb aller Wumpus-Agenten der verschiedenen Praktikumsgruppen durchführen. Bis dahin stellen wir euch unseren Agenten *wumpus-agent-aa* zur Verfügung, gegen den ihr euren Agenten testen könnt.

Die Funktion `agent-trials` erhält eine Wumpus-Welt, eine Liste der zu testenden Agenten und die Anzahl der Läufe.

```
(agent-trials 'make-wumpus-world
  '(wumpus-agent-pxx wumpus-agent-aa aimless-wumpus-agent random-wumpus-agent)
  :n 10)
```

Die Funktion hat außerdem einen Key-Parameter `:show`. Ist dieser auf T gesetzt (so ist es standardmäßig), dann werden die einzelnen Test-Welten ausgegeben.

### **Aufgabe 1:**

Zu schreiben ist eine modellbasierte Agentenfunktion für die Wumpus-Welt. Diese ist unbedingt mit der Funktion *agent-trials* zu testen! Nur durch den mehrmaligen Durchlauf kann geprüft werden, ob alle Werte vor/nach einem Lauf korrekt zurückgesetzt werden und nur dann kann der Wettbewerb reibungslos ablaufen.

### **Weiterer Hinweis:**

- Damit euer Programm auch in größeren Wumpus-Welten funktioniert, solltet ihr die Größe der Wumpus-Welt immer mit der globalen Variablen `gridSize*` bestimmen.