



## Artificial Intelligence, Wintersemester 2012/2013

### Übungsblatt 2

Abgabe: 25.10.12, Besprechung: 8.11.12

#### **Aufgabe 1 Umgebungen für Agentensysteme 1 [4 Punkte]**

Bestimmen Sie für jede der folgenden Umgebungen die sechs in der Vorlesung genannten Eigenschaften: Observable, Deterministic, Episodic, Static, Discrete, # of Agents.

- (a) Mensch ärgere dich nicht
- (b) Skat
- (c) Billard
- (d) Fußball

#### **Aufgabe 2 Umgebungen für Agentensysteme 2 [4 Punkte]**

Geben Sie je mindestens zwei Szenarien für die folgenden Eigenschaftskombinationen an. Die Antworten dürfen sich nicht wiederholen und weder in der Vorlesung noch in vorherigen Aufgaben genannt worden sein.

- (a) vollständig beobachtbar, diskret, episodisch
- (b) kontinuierlich, partiell beobachtbar, dynamisch
- (c) mehrere Agenten, dynamisch
- (d) mehrere Agenten, nicht-deterministisch, diskret

#### **Aufgabe 3 Rationalität 1 [4 Punkte]**

- (a) Ein Agent muss entscheiden, ob er ein Spiel spielen soll. Sein Einsatz beträgt 1 Euro. Es werden zwei 6-seitige Würfel geworfen. Wenn keiner der Würfel die 6 zeigt, bekommt der Agent kein Geld zurück ( $-1$  Euro). Erscheint die 6 auf nur einem der Würfel, bekommt er 3 Euro ( $+2$  Euro). Wenn zwei Würfel die 6 zeigen, bekommt er 5 Euro ( $+4$  Euro). Der Agent will seinen erwarteten Geldgewinn maximieren. Ist es rational, wenn der Agent auf das Spiel eingeht? Begründen Sie Ihre Antwort.
- (b) Bei einer Variante des Spiels werden insgesamt drei 6-seitige Würfel geworfen. Die Regeln für Gewinne sind die gleichen, aber wenn alle drei Würfel die 6 zeigen, bekommt der Agent ebenfalls kein Geld zurück ( $-1$  Euro). Ist es rational, wenn der Agent auf diese Variante des Spiels eingeht? Begründen Sie Ihre Antwort.

#### **Aufgabe 4 Rationalität 2 [3 Punkte]**

Bestimmen Sie den einfachsten Agenten (Simple Reflex Agent, Model-based Reflex Agent, Model-based Goal-based Agent, Model-based Utility-based Agent, oder Learning Agent), der für die folgenden Situationen notwendig ist, und begründen Sie Ihre Antwort. Beachten Sie, dass Ihre Lösungen praktisch umsetzbar sein müssen. Diskutieren Sie eventuell Grenzfälle.

- (a) Für eine gegebene Menge von Piloten und Flugzeugen, welche bestimmten Einsatzregeln unterliegen, muss der Agent einen Flugplan finden, der keine der Regeln verletzt.
- (b) Der Agent muss Tic Tac Toe spielen.
- (c) Ein Musikempfehlungsdienst soll seinen Nutzern basierend auf deren Hörverhalten möglichst gute Vorschläge machen, welche anderen Musikstücke ihnen auch gefallen könnten. Dabei müssen auch neu veröffentlichte Musikstücke beachtet werden.

#### **Aufgabe 5 LISP: Türme von Hanoi [5 Punkte]**

In dem Spiel „Türme von Hanoi“ gibt es drei Stapel  $A$ ,  $B$  und  $C$ , auf die mehrere, verschieden große Scheiben gelegt werden. Es darf aber niemals eine größere Scheibe auf einer kleineren Scheibe liegen, d. h. zu jedem Zeitpunkt sind auf jedem Stapel die Scheiben von unten nach oben in absteigender Größe sortiert.

Zu Beginn des Spiel liegen alle Scheiben geordnet auf Stapel  $A$ . Ziel des Spiels ist es, diesen gesamten Turm auf Stapel  $C$  zu verschieben, wobei in jedem Zug nur genau eine Scheibe unter Beachtung der Größensortierung bewegt werden darf.

- (a) Implementieren Sie ein Programm in LISP, welches diese Aufgabe löst. Laden Sie sich von der Übungsseite die Datei *hanoi.lsp* herunter. In dieser finden Sie einen vordefinierten Funktionsrumpf für *move-helper* sowie die Funktionen (*move x y stacks*) und (*hanoi-append l e*). Die drei Stapel sind gespeichert in der Liste *stacks*. Jeder Stapel ist selbst auch als Liste dargestellt, wobei das erste Element der untersten Scheibe entspricht. (*move x y stacks*) soll nun den Stapel mit Index  $x$  auf den Stapel mit Index  $y$  verschieben, wobei der Zustand aller Stapel in *stacks* gespeichert ist. Dabei ruft sie die Hilfsfunktion (*move-helper n x y stacks*) auf, welche die obersten  $n$  Scheiben von  $x$  nach  $y$  verschieben soll. Vervollständigen Sie nun diese Hilfsfunktion an der dafür vorgesehenen, markierten Stelle. Um eine Scheibe auf einen Stapel zu legen, verwenden Sie die Methode (*hanoi-append l e*), welche ein Liste zurückgibt, bei der das Element  $e$  an die Liste  $l$  unter Beachtung der Größensortierung angehängt wurde. Falls  $e$  größer als das letzte Element der Liste ist, gibt die Funktion *NIL* zurück.
- (b) Erweitern Sie das Programm so, dass es unter Beachtung der Spielregeln ausgehend von einer beliebigen gültigen Belegung der Stapel alle Scheiben sortiert auf einen beliebigen Stapel verlagert.

Senden Sie Ihre kommentierte Implementierung sowie Ihre Ergebnisse an [andreas.draeger@uni-tuebingen.de](mailto:andreas.draeger@uni-tuebingen.de) UND [florian.mittag@uni-tuebingen.de](mailto:florian.mittag@uni-tuebingen.de) mit dem Betreff „Abgabe KI-Uebung 02“.