



Universität Koblenz-Landau  
Fachbereich Informatik

**Klausur**

***Einführung in die Künstliche Intelligenz /  
Künstliche Intelligenz für IM***

SS 2003

Dr. B. Beckert

2. August 2003

**Name, Vorname:** \_\_\_\_\_

**Matrikel-Nr.:** \_\_\_\_\_

**Studiengang:**  Informationsmanagement (Bachelor)  
 Informatik (Diplom)  
 Anderer: \_\_\_\_\_

**Freiversuch:**  ja  nein **Wenn Sie hier nichts ankreuzen,  
gilt das als „nein“**

A1 (10)	A2 (17)	A3 (15)	A4 (6)	A5 (12)	Σ (60)

**Bewertungstabelle bitte frei lassen !!!**

**Zum Bestehen der Klausur benötigen Sie 20 der erreichbaren 60 Punkte.**

**Wichtiger Hinweis:**

**Bei Ankreuzaufgaben wird für falsche Kreuze ein Punkt abgezogen!**

**Dabei werden insgesamt jedoch keinesfalls weniger als 0 Punkte für die jeweilige Teilaufgabe vergeben.**

**1 Agenten (4 + 6 Punkte)**

a) (4 Punkte)

Entscheiden Sie durch Ankreuzen, ob die folgenden Aussagen richtig oder falsch sind.

Rationales Verhalten setzt voraus, daß der Agent Bewusstsein und eine Seele besitzt.	richtig <input type="checkbox"/> falsch <input checked="" type="checkbox"/>
Rationale Agenten führen in jeder Situation immer die Aktion aus, die unter allen möglichen Aktionen die Gewinnfunktion ( <i>performance measure</i> ) maximiert.	richtig <input type="checkbox"/> falsch <input checked="" type="checkbox"/>
Ein diskretes Environment kann unendlich viele Zustände haben.	richtig <input checked="" type="checkbox"/> falsch <input type="checkbox"/>
Ein Agent ist durch eine Funktion beschreibbar, welche eine Reihe von Wahrnehmungen des Agenten auf die Menge seiner möglichen Aktionen abbildet.	richtig <input checked="" type="checkbox"/> falsch <input type="checkbox"/>

b) (6 Punkte)

Es soll ein Agent implementiert werden, der Billiard spielen kann. Der Agent soll auf einem realen Billardtisch mittels einer Videokamera und eines Roboterarms gegen einen menschlichen Gegner spielen können.

Kreuzen Sie die Eigenschaften an, die das Environment des Billiard-Agent hat.

fully observable	<input checked="" type="checkbox"/>
partially observable	<input type="checkbox"/>
deterministic	<input type="checkbox"/>
strategic	<input checked="" type="checkbox"/>
stochastic	<input checked="" type="checkbox"/>
episodic	<input type="checkbox"/>
sequential	<input checked="" type="checkbox"/>
static	<input checked="" type="checkbox"/>
dynamic	<input type="checkbox"/>
semi-dynamic	<input type="checkbox"/>
discrete	<input type="checkbox"/>
continuous	<input checked="" type="checkbox"/>
single agent	<input type="checkbox"/>
multi agent	<input checked="" type="checkbox"/>

**Lösung:**

Da man den stochastischen Anteil am Ergebnis eines Billardstoßes verschieden hoch bewerten kann, gilt es als richtige Antwort, wenn die Eigenschaften „strategisch“ und „stochastisch“ beide angekreuzt sind wie auch wenn nur eine von beiden angekreuzt ist.

## 2 Suchen (5 + 2 + 10 Punkte)

a) (5 Punkte)

Entscheiden Sie durch Ankreuzen, ob die folgenden Aussagen richtig oder falsch sind.

In einem endlichen Suchraum besucht Breitensuche im schlechtesten Fall ( <i>worst case</i> ) genauso viele Knoten wie Tiefensuche im schlechtesten Fall besucht.	richtig <input checked="" type="checkbox"/> falsch <input type="checkbox"/>
<i>Iterative-deepening search</i> ist in praktischen KI-Anwendungen Breiten- und Tiefensuche zumeist überlegen.	richtig <input checked="" type="checkbox"/> falsch <input type="checkbox"/>
Die erste Lösung, die man mit $A^*$ -Suche findet, ist eine optimale Lösung.	richtig <input checked="" type="checkbox"/> falsch <input type="checkbox"/>
Gegeben seien zwei Heuristikfunktionen $h_1$ und $h_2$ , so dass $h_1$ die Funktion $h_2$ dominiert. Dann ist die erste Lösung, die $A^*$ -Suche mit $h_2$ findet, immer schlechter als die erste Lösung, die $A^*$ -Suche mit $h_1$ findet.	richtig <input type="checkbox"/> falsch <input checked="" type="checkbox"/>
$\alpha$ - $\beta$ -Pruning erlaubt die Suchtiefe (bei gleich bleibendem Aufwand) in etwa zu verdoppeln.	richtig <input checked="" type="checkbox"/> falsch <input type="checkbox"/>

b) (2 Punkte)

Entscheiden Sie durch Ankreuzen:

Wir betrachten den Aufwand von Breitensuche in einem Suchbaum mit Verzweigungsgrad 2. Wenn der Abstand zur Wurzel der am weitesten oben liegenden Lösung verfünffacht wird, dann steigt der Suchaufwand (d.h., die Zahl der besuchten Knoten bis eine Lösung gefunden wird) ungefähr um den Faktor

- 1   
 5   
 10   
 25   
 32

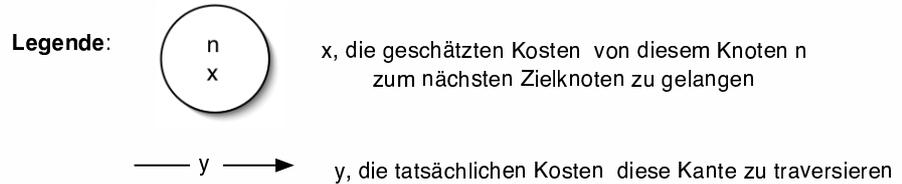
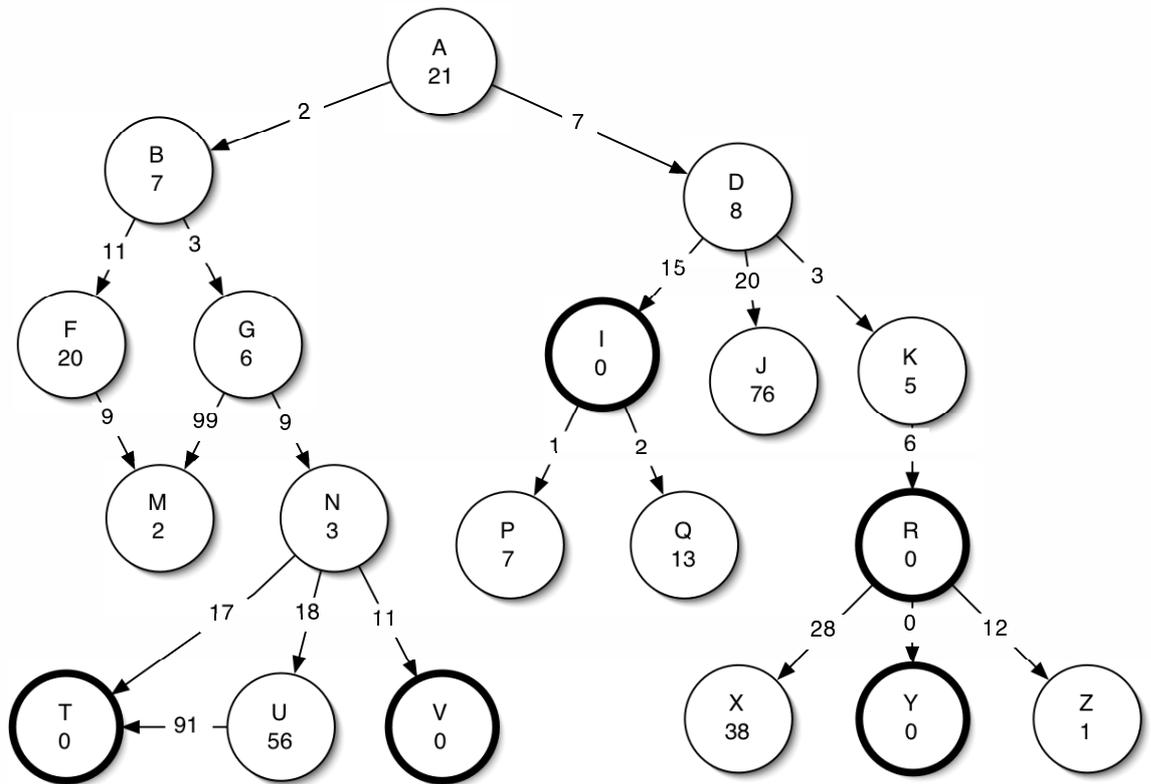
c) (10 Punkte)

Betrachten Sie den folgenden Suchbaum. Der initiale Zustand ist der Knoten A (ganz oben). Zielzustände sind durch Knoten mit dickerer Umrandung repräsentiert.

Geben Sie – für jede der unten aufgeführten Suchstrategien – den zuerst gefundenen Zielzustand und die Reihenfolge an, in der die Knoten expandiert werden.

**Hinweis:**

- Ein Knoten gilt als expandiert, wenn er als Zielzustand erkannt bzw. seine Kindknoten zur Liste der noch zu untersuchenden Knoten hinzugefügt worden sind.
- Bei Tiefen- und Breitensuche werden Kindknoten von links nach rechts selektiert.



**Tiefensuche**

Erreichter Zielzustand: T Reihenfolge: A,B,F,M,G,(M,)N,T  
Lösung ist mit und ohne den zweiten Besuch von M richtig

**Breitensuche**

Erreichter Zielzustand: I Reihenfolge: A,B,D,F,G,I

**Uniform-Cost-Suche**

Erreichter Zielzustand: R Reihenfolge: A,B,G,D,K,F,N,R

**A\*-Suche**

Erreichter Zielzustand: R Reihenfolge: A,B,G,D,K,R

### 3 Aussagenlogik (4 + 1 + 10 Punkte)

a) (4 Punkte)

Entscheiden Sie durch Ankreuzen, ob die folgenden Aussagen richtig oder falsch sind.

Es gibt Beweisverfahren für die Aussagenlogik (Kalküle), die korrekt aber nicht vollständig sind	richtig <input checked="" type="checkbox"/> falsch <input type="checkbox"/>
Resolution ist ein korrekter und vollständiger Kalkül, um die Unerfüllbarkeit aussagenlogischer Formeln zu prüfen	richtig <input checked="" type="checkbox"/> falsch <input type="checkbox"/>
Jede aussagenlogische Formel, die erfüllbar ( <i>satisfiable</i> ) ist, ist auch allgemeingültig ( <i>valid</i> ).	richtig <input type="checkbox"/> falsch <input checked="" type="checkbox"/>
Folgende aussagenlogische Formel ist unerfüllbar: $\neg((A \wedge B) \Rightarrow A)$	richtig <input checked="" type="checkbox"/> falsch <input type="checkbox"/>

b) (1 Punkt)

Kreuzen Sie diejenige der folgenden Formeln an, die in konjunktiver Normalform ist:

$$p \vee (q \wedge r) \quad \square$$

$$p \wedge (q \vee r) \quad \boxed{\times}$$

c) (10 Punkte)

Gegeben sei folgende aussagenlogische Formel  $Fml$ :

$$\neg(\neg A \wedge \neg B) \wedge (A \Rightarrow C) \wedge (\neg B \vee D) \wedge (E \vee (\neg C \wedge \neg D))$$

Formen Sie die Formel  $Fml$  zunächst in konjunktive Normalform um (5 Punkte).

Zeigen Sie dann mittels des Resolutionsverfahrens, dass  $E$  aus der Formel  $Fml$  logisch folgt, d.h., dass  $Fml \models E$  gilt (5 Punkte).

### Lösung:

#### Umformung in konjunktive Normalform

Die Formel  $Fml$  hat vier konjunktiv verknüpfte Teilformeln. Diese können einzeln in (Konjunktionen von) Diskunktionen umgeformt werden:

$$\begin{aligned}\neg(\neg A \wedge \neg B) &\equiv A \vee B \\ A \Rightarrow C &\equiv \neg A \vee C \\ \neg B \vee D &\equiv \neg B \vee D \\ E \vee (\neg C \wedge \neg D) &\equiv (E \vee \neg C) \wedge (E \vee \neg D)\end{aligned}$$

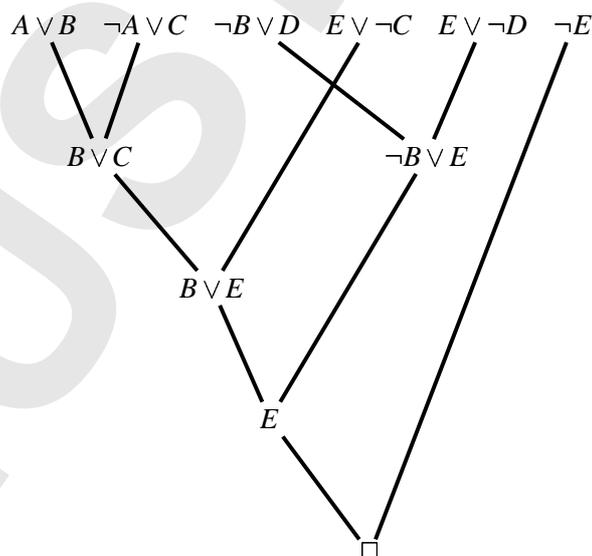
Insgesamt ergibt dies als konjunktive Normalform von  $Fml$ :

$$(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (E \vee \neg C) \wedge (E \vee \neg D)$$

#### Resolutionsbeweis

Beim Resolutionsbeweis für  $Fml \models E$  hat man als Klauseln die Disjunktionen aus der zuvor berechneten konjunktive Normalform von  $Fml$  sowie das Negat von  $E$ , also  $\neg E$ , zur Verfügung.

Resolutionsbeweis:



(Es gibt noch viele andere Beweise. Dies ist nur eine der Möglichkeiten.)

#### 4 Prädikatenlogik (2 + 2 + 2 Punkte)

a) (2 Punkte)

Entscheiden Sie durch Ankreuzen, ob die folgenden Aussagen richtig oder falsch sind.

Für alle prädikatenlogischen Formeln $A$ gilt: Falls $A$ erfüllbar ( <i>satisfiable</i> ) ist, dann ist $\neg A$ unerfüllbar ( <i>unsatisfiable</i> ).	richtig <input type="checkbox"/> falsch <input checked="" type="checkbox"/>
Für alle prädikatenlogischen Formeln $A$ gilt: Falls $A$ unerfüllbar ( <i>unsatisfiable</i> ) ist, dann ist $\neg A$ allgemeingültig ( <i>valid</i> ).	richtig <input checked="" type="checkbox"/> falsch <input type="checkbox"/>

b) (2 Punkte)

Entscheiden Sie durch Ankreuzen, ob die folgenden prädikatenlogische Formeln korrekte Formalisierungen des dazugehörigen natürlichsprachlichen Satzes sind.

Es gibt keine zwei Personen, die dieselbe Sozialversicherungsnummer haben. $\neg \exists x, y, n ((person(x) \wedge person(y)) \Rightarrow (hasNum(x, n) \wedge hasNum(y, n)))$	richtig <input type="checkbox"/> falsch <input checked="" type="checkbox"/>
Die Sozialversicherungsnummer jeder Person hat 9 Stellen. $\forall x, n ((person(x) \wedge hasNum(x, n)) \Rightarrow digits(n, 9))$	richtig <input checked="" type="checkbox"/> falsch <input type="checkbox"/>

c) (2 Punkte)

Geben Sie eine prädikatenlogische Formel an, die die folgende Aussage formalisiert:

*Jeder Taschenrechner hat mindestens eine Batterie.*

**Hinweis:**

Verwenden Sie dazu die einstelligen Prädikate *calculator* und *battery* sowie das zweistellige Prädikat *isPartOf*.

**Lösung:**

$$\forall x (calculator(x) \Rightarrow \exists y (battery(y) \wedge isPartOf(y, x)))$$

## 5 Planen (10 + 2 Punkte)

Die möglichen Aktionen eines Roboters seien durch folgende drei STRIPS-Operatoren beschrieben:

<i>Action:</i>	$pick(o)$
<i>Precond:</i>	$at(Robot, x) \wedge at(o, x)$
<i>Effect:</i>	$\neg at(o, x) \wedge holding(o)$

<i>Action:</i>	$drop(o)$
<i>Precond:</i>	$at(Robot, x) \wedge holding(o)$
<i>Effect:</i>	$at(o, x) \wedge \neg holding(o)$

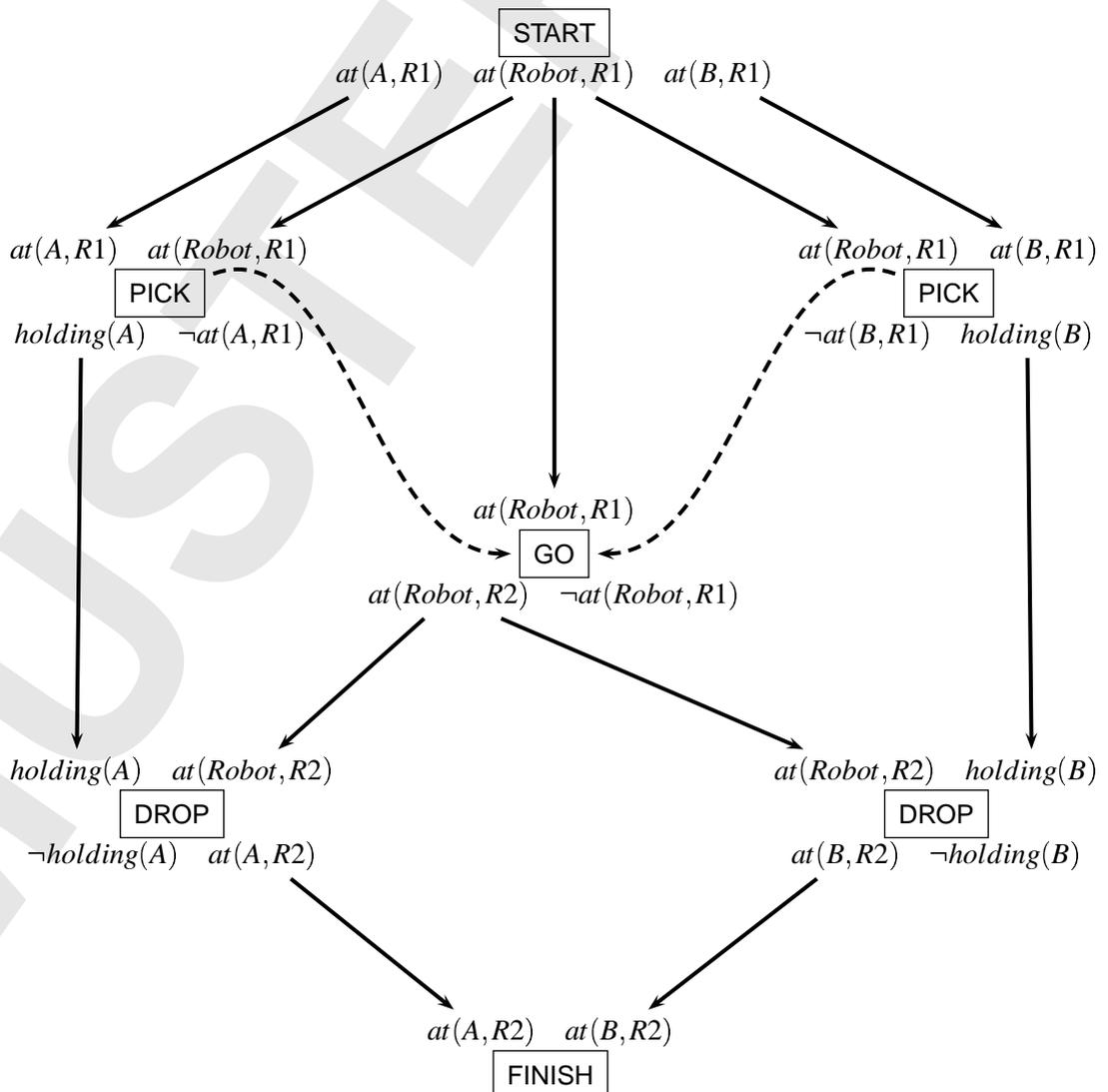
<i>Action:</i>	$go(x, y)$
<i>Precond:</i>	$at(Robot, x)$
<i>Effect:</i>	$\neg at(Robot, x) \wedge at(Robot, y)$

a) (10 Punkte)

Verwenden Sie den POP-Algorithmus, um den unten dargestellten partiellen Plan zu vervollständigen (der Roboter soll die Teile A und B von R1 nach R2 bringen, wobei er beide Teile gleichzeitig halten kann).

### Hinweis:

Zeichnen Sie neben den notwendigen zusätzlichen Aktionen auch deren Vorbedingungen (*preconditions*) und Effekte sowie die kausalen Links und die zur Beseitigung von Clobberings notwendigen Ordnungsconstraints (*ordering constraints*) ein.



b) (2 Punkte)

Mit den gegebenen Operatoren ist es möglich, dass der Roboter mehr als ein Objekt gleichzeitig hält.

Modifizieren Sie die Operatoren *pick* und *drop* so, dass der Roboter nur ein Objekt gleichzeitig halten kann. Verwenden Sie dazu das zusätzliche Prädikat *EmptyHand(x)*.

**Lösung:**

*Action:*    *pick(o)*  
*Precond:*    $at(Robot, x) \wedge at(o, x) \wedge EmptyHand(Robot)$   
*Effect:*      $\neg at(o, x) \wedge holding(o) \wedge \neg EmptyHand(Robot)$

*Action:*    *drop(o)*  
*Precond:*    $at(Robot, x) \wedge holding(o)$   
*Effect:*      $at(o, x) \wedge \neg holding(o) \wedge EmptyHand(Robot)$