Abschlußklausur **19. Juli 2002**

Grundlagen der Künstlichen Intelligenz

Albert-Ludwigs Universität Freiburg
Institut für Informatik

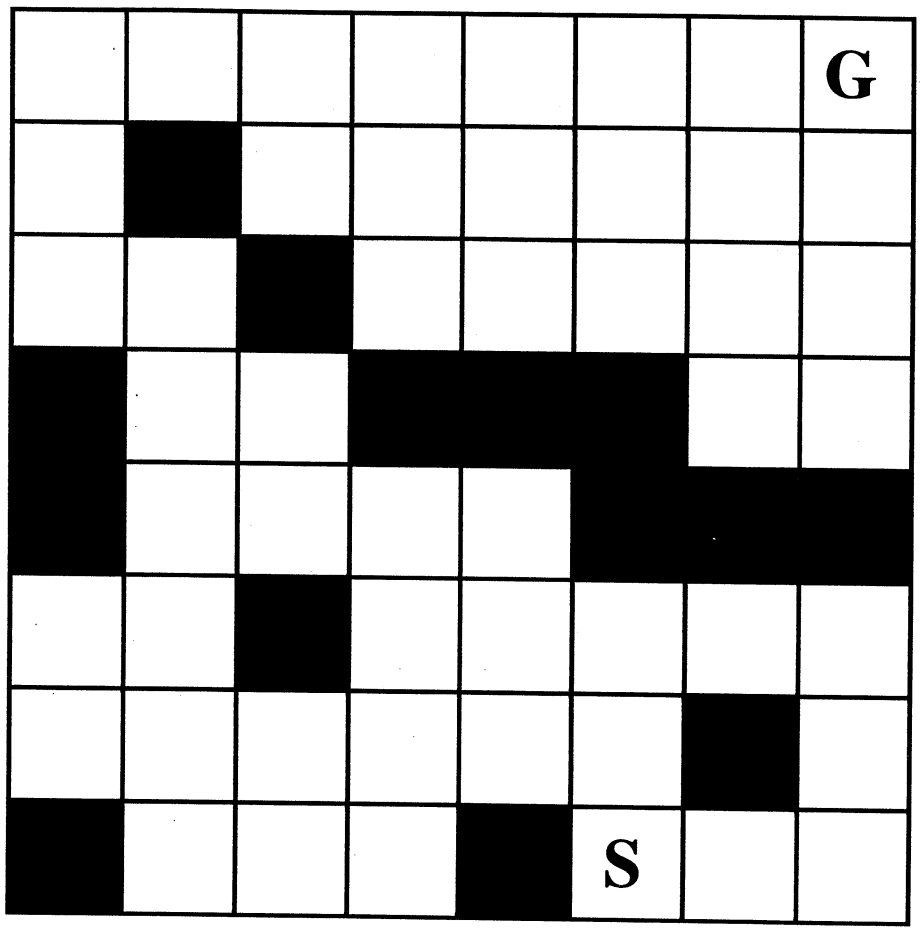
Die Klausur besteht aus **12 Frage**. Insgesamt sind **93 Punkte** zu erreichen.

Die Klausur dauert **90** Minuten. Es ist nicht erlaubt Bücher, Skripte oder ähnliche Unterlagen zu benutzen. Sollte der Platz für eine Antwort nicht ausreichen, so schreiben Sie bitte auf der Rückseite der jeweiligen Seite weiter. Bitte vermerken Sie auf jeder einzelnen Seite an dem dafür vorgesehenen Platz Ihre **Matrikelnummer**.

Nachname, Vorname: _____

Matrikelnummer: _____

1. (10 Punkte) Das Problem der Wegplanung in *Grid-Worlds*, wie sie weiter unten eine sehen, besteht darin, einen Pfad von einem Feld S zu einem Feld G zu finden. Ein Agent kann jeweils *ein* Feld in horizontaler oder vertikaler Richtung gehen. Schwarze Felder können nicht betreten werden.



Nehmen Sie an, dass der Agent *Bidirectional* sucht, um einen Pfad von S nach G zu finden. Genauer gesagt, benutzt er eine *Depth-First Search* von S ausgehend und eine *Breath-First Seach* von G ausgehen. In beiden Fällen ist the Reihenfolge der Operatoren *rauf*, *rechts*, *runter* und *links*. Zyklen werden vermieden. Der Agent beginnt bei S und führt abwechselnd eine *Dept-First search* Operation und eine *Breath-First Search* Operation aus. Nummerieren Sie in der gegebenen *Grid-World* für beide Suchtechnicken die Felder in der Reihenfolge, in der sie expandiert werden. Hören sie auf, sobald eine Lösung gefunden worden ist, und markieren Sie die Lösung.

2. Das folgende Puzzle soll mit Hilfe des A^* Algorithmus gelöst werden. Gegenstand des Puzzles sind Zahlen zwischen 100 und 999. Anfangs sind zwei Zahlen S und G gegeben, sowie eine Menge *Bad* von Zahlen. Ein Spielzug besteht darin, eine Zahl in eine andere Zahl zu verwandeln, indem man 1 zu einer Ziffer der Zahl addiert oder 1 von einer Ziffer subtrahiert. Ein gültiger Zug wäre also beispielsweise von 678 nach 679 oder von 234 nach 134. Jeder Zug hat die Kosten 1. Zusätzlich unterliegen die Züge den folgenden Einschränkungen:

   - Es ist nicht erlaubt, zu der Ziffer 9 zu addieren oder von der Ziffer 0 zu subtrahieren.
   - Es ist nicht erlaubt, einen Zug auszuführen, der die derzeitige Zahl in eine Zahl aus der Menge *Bad* überführt.
   - Ein Spieler darf dieselbe Ziffer nicht in zwei aufeinanderfolgenden Zügen ändern.

   Lösen Sie das Puzzle, indem Sie von S nach G mit der kleinstmöglichen Anzahl an Zügen gelangen.

   (a) (1 Punkte) Geben Sie eine *State Description* (Zustandsbeschreibung) an, mit der der A^* Algorithmus angewendet werden kann.

   (b) (3 Punkte) Definieren Sie eine *admissible Heuristik*, die für dieses Problem in einer A^* Suche verwendet werden kann. Erläutern Sie, warum Ihre Heuristik admissible ist.

   (c) (6 Punkte) Verwenden Sie die Heuristik aus (b), um die ersten drei (3) *Knoten-Expansionen* der A^* Suche für $S = 567$, $G = 777$ und $Bad = \{666, 667\}$ durchzuführen. Kennzeichnen Sie in dem Baum auch alle legalen Nachfolger von jedem Knoten, den Sie expandieren.

3. (5 Punkte) Finden Sie mittels der *Davis-Putnam* Prozedur ein Model der folgenden aussagenlogischen Formel, oder beweisen Sie, dass die Formel unerfüllbar ist.

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

4. Betrachten Sie die folgenden beiden Sätze in Prädikatenlogik erster Stufe.

   1. $\phi_1 \equiv \forall x(boy(x) \Rightarrow \exists y(girl(y) \wedge likes(x,y)))$
   2. $\phi_2 \equiv \exists y(girl(y) \wedge \forall x(boy(x) \Rightarrow likes(x,y)))$

   (a) (3 Punkte) Welche Folgerung ist korrekt? Tragen Sie "ja" oder "nein" in die entsprechenden Kästchen ein. **(-2 für eine falsche Antwort)**

   1. ☐ $\phi_1 \models \phi_2$
   2. ☐ $\phi_2 \models \phi_1$

   (b) (10 Punkte) Beweisen Sie Ihre Behauptung mittels *Resolution Refutation.*

5. (5 Punkte) Geben Sie ein Prolog-Program an, dass `append(X, Y, Z)` so definiert, dass `Z` die Konkatenation der Listen `X` und `Y` ist. Nehmen Sie an, dass alle drei Listen `Z`, `Y` und `X` als *Difference Lists* repräsentiert sind.

6. Betrachten Sie das folgende Prolog-Programm:

```
part(a). part(b). part(c). part(d).
red(a). black(b). red(c). black(c).
color(P,red) :- red(P).
color(P,black) :- black(P).
color(P,unknown).
```

   (a) (4 Punkte) Zeichnen Sie den SLD-Baum für die Anfrage `?- color(a, C).`

(b) (3 Punkte) In dem Programm wird die Klausel `color(P,red) :- red(P)` durch

```
color(P,red) :- red(P),!.
```

ersetzt. Markieren Sie im SLD-Baum, welche Zweige dadurch abgeschnitten werden.

(c) (3 Punkte) Ändert dieser Cut die Semantik des Programmes? Falls ja, erklären Sie kurz, warum. **(-3 Punkte für eine falsche Antwort)**

7. (5 Punkte) Zeichen sie den für *phase transitions* typischen Graphen, und erklären Sie anhand des Graphens in höchstens 5 Sätzen, was eine *phase transition* ist.

8. (8 Punkte) Es sei folgendes *Constraint Satisfaction Problem* (CSP) gegeben. Über der Menge $\{A, E, G, K, M, O, R, S, T\}$ von 9 Bool'schen Variablen, d.h. alle Variablen haben die Domäne $\{true, false\}$, sind die folgenden *(Primitive) Constraints* gegeben:

$$\begin{aligned} M &\Leftrightarrow A \\ G \wedge O &\Leftrightarrow R \\ E \wedge R &\Leftrightarrow K \\ T \wedge E &\Leftrightarrow S \end{aligned}$$

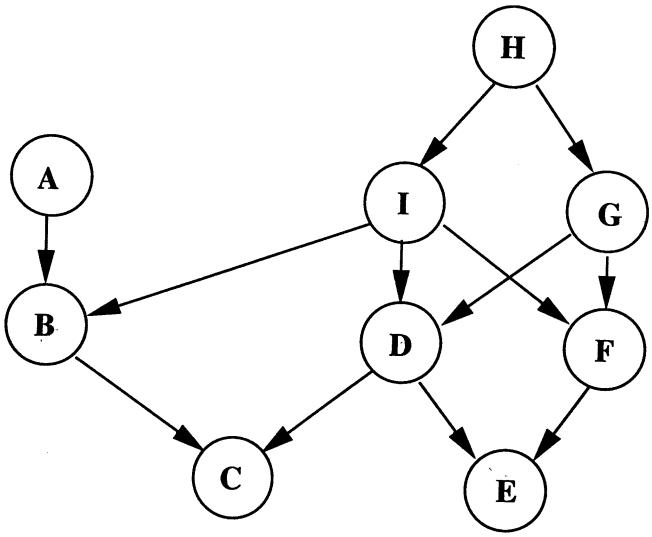
Nehmen Sie an, dass ein Algorithmus zum Lösen von CSPs schon die folgenden Variablenbelegungen getroffen hat:

$$A = false, G = true, K = true, M = true, O = true.$$

Geben Sie für jedes *(Primitive) Constraint* an, ob es *arc consistent* (AC) und/or *node consistent* (NC) ist gegeben die selektierten Variablenbelegungen. Dazu tragen Sie "ja" oder "nein" in die entsprechenden Kästchen ein **(-1 Punkt für jeden falschen Eintrag)**:

1. NC[ ] AC[ ] $M \Leftrightarrow A$
2. NC[ ] AC[ ] $G \wedge O \Leftrightarrow R$
3. NC[ ] AC[ ] $E \wedge R \Leftrightarrow K$
4. NC[ ] AC[ ] $T \wedge E \Leftrightarrow S$

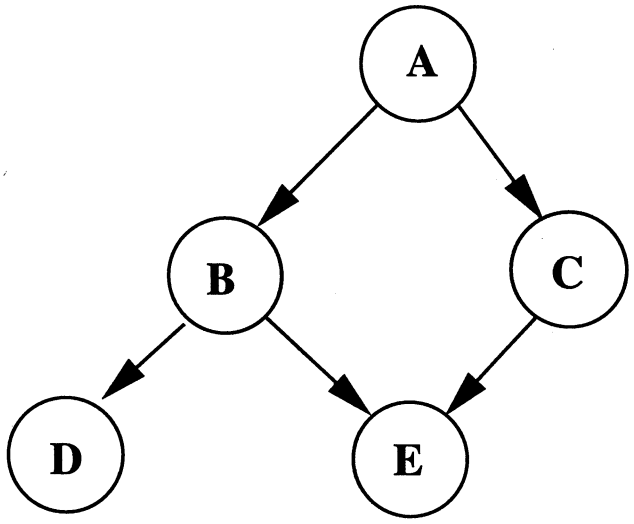
9. Betrachten Sie folgendes *Bayesian Network*:



(a) (3 Punkte) D-separiert die Menge $\{B, E, H\}$ den Knoten A von G? Geben Sie als Antwort **ja** oder **nein** an. Im Falle von **nein** geben sie einen Pfad an, der nicht blockiert ist. **(-2 Punkte für eine falsche Antwort.)**

(b) (3 Punkte) D-separiert die Menge $\{I\}$ den Knoten B von E? Geben Sie als Antwort **ja** oder **nein** an. Im Falle von **nein** geben sie einen Pfad an, der nicht blockiert ist. **(-2 Punkte für eine falsche Antwort.)**

10. (6 Punkte) Gegeben sei das folgende *Bayesian Network*:



Zeigen Sie, wie *Variable Elimination* die Wahrscheinlichkeit von $E = e$ berechnet bei Anwendung der *Elimination Order* A, B, C, D.

11. (10 Punkte) In dieser Aufgaben soll ein Konzept erlernt werden. Die Instanzen sind durch die Bool'schen Variablen A und B beschrieben. Es seien die folgenden zwei Beispiele gegeben:

A	B	Class
true	true	positive
false	false	negative

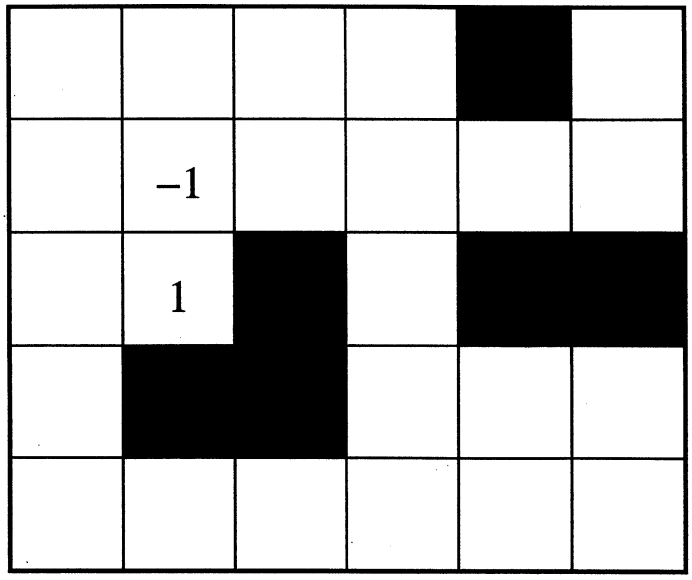
Nehmen Sie an, dass der Hypothesenraum H aus allen möglichen *Disjunktionen* über den Variablen A und B besteht, d.h.

$$H =_{def} \{ \; true, \; A, \neg A, B, \neg B, \; A \vee B, A \vee \neg B, \neg A \vee B, \neg A \vee \neg B, \; false \; \}$$

wobei $A \vee B$ für $(A = true \vee B = true)$ steht. Die zwei weiter oben angebenen Beispiele induzieren einen *version space* in H. Geben sie die *G-set* und die *S-set*, die den *version space* beschreiben.

Matrikelnummer:

12. (5 Punkte) Betrachten Sie die deterministiche Version der folgenden *Grid-World*:



Wie üblich kann der Roboter sich immer nur ein Feld nach *oben*, *links*, *rechts* oder *unten* bewegen. Geben Sie im Diagramm die *optimal policy* an, wobei sich der Nutzen (*utility*) einer Aktionssequenz als

$$\textit{Endwert} - \frac{1}{100} \cdot \textit{Anzahl der Aktionen.}$$

berechnet.