

# 4. Übung zur Vorlesung Künstliche Intelligenz

Institut für Informatik, FU Berlin, SoSe 2006  
Prof. Dr. Raúl Rojas, Marco Block, Ernesto Tapia

Alle Übungsaufgaben sollen mit SWI-Prolog gelöst werden. Neben der schriftlichen Abgabe, sind die Programmieraufgaben **zusätzlich** per e-mail an den Tutor zu schicken. **Eine e-mail ersetzt nicht die schriftliche Abgabe!** Zur Erinnerung: Testläufe gehören zur Abgabe und werden ebenfalls bewertet.

Aus gegebenem Anlass ist es notwendig, auf diesem Übungszettel folgende Daten aufzuführen:

- Nachname, Vorname
- Matrikelnummer
- Studiengang (Bachelor, Master, Diplom) (← sehr wichtig!)
- Fachsemesterzahl

## 1. Aufgabe (2 Punkte) *Tiefensuche*

Sie kennen die 3 Bearbeitungsstrategien in der Tiefensuche. *Preorder*, *Inorder* und *Postorder*. Geben Sie die 2 noch fehlenden Varianten der Tiefensuche an, bei denen die Knoten in der entsprechenden Reihenfolge ausgegeben werden. Wenn es Schwierigkeiten bei einer der beiden fehlenden Varianten gibt, so beschreiben Sie das Problem und erörtern Sie gegebenenfalls, warum es nicht funktionieren kann.

## 2. Aufgabe (6 Punkte)

In der Vorlesung wurde das **8-Puzzle-Spiel** vorgestellt. Es gab eine Zielkonfiguration:

1	2	3
8	<b>b</b>	4
7	6	5

Um nicht alle Kombinationen aufschreiben zu müssen, wurde eine erste Optimierung besprochen, die nur 24 Prädikate benötigt. Überlegen Sie sich eine Strategie, um die Anzahl der Prädikate auf 6 zu reduzieren, indem Sie zusätzlich eine Rotation erlauben.

- Geben Sie die 6 Prädikate + Rotation an.
- Prüfen Sie Ihre Prädikate mit folgenden Eingaben, indem Sie die Tiefensuche verwenden:

1	8	7	1	6	7	8	7	5	3	2	6	1	2	3
2	<b>b</b>	6	<b>b</b>	2	8	2	3	1	8	1	4	7	8	4
3	4	5	5	4	3	4	<b>b</b>	6	<b>b</b>	7	5	6	<b>b</b>	5

Wieviele Knoten besitzt Ihr Suchbaum jeweils?

- c) Machen Sie sich Gedanken über eine weitere Optimierung, um die Komplexität des Suchbaumes zu reduzieren. Wie könnte eine Bewertungsfunktion helfen und wie müsste diese aussehen?

### 3.Aufgabe (6 Punkte)

Es gibt eine Variante der Tiefensuche, die sich auf eine bestimmte maximale Länge beschränkt.

- a) Geben Sie ein Prädikat **maxdepth** an, dass die Knoten des Suchbaumes wie bei der Tiefensuche abarbeitet, aber den Suchbaum auf eine vorgegebenen Tiefe  $t$  beschränkt. Sollte keine Lösung bis zu dieser Tiefe gefunden werden, so behandeln wir diesen Knoten wie ein **fail**.
- b) Schreiben Sie ein Prädikat, dass die Beispiele aus 2 b) mit  $t = 1, 2, 4, 8$  iterativ löst. Zu Beginn wird ein Suchbaum der Tiefe 1 geprüft, usw. Sollte bei einer bestimmten Tiefe eine Lösung gefunden worden sein, so bricht die Suche ab.

**Abgabe:** (Freitag) 26.05.2006, 12:00 Uhr (s.t.)  
(verspätete Abgaben werden nicht mehr entgegen genommen!)