

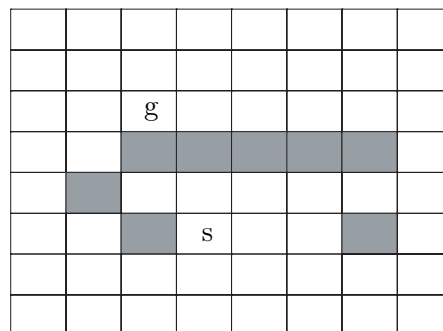
# GWV – Grundlagen der Wissensverarbeitung

## Aufgabenzettel 3: Suche / Regelbasierte Systeme (II)

Abgabe 16/17.11.2008 Besprechung am 17/18.11.2008.

### Übungsaufgabe 3.1: (Suche (WBS))

von
8



In der oben abgebildeten Umgebung befindet sich an der mit „s“ gekennzeichneten Stelle ein Gegenstand. Die Aufgabe ist, einen Pfad von s nach g zu finden. Der Gegenstand darf um genau ein Feld in horizontaler oder vertikaler Richtung bewegt werden. Die dunklen Kästchen sind verbotene Felder.

(Zur Lösung der Teilaufgaben Aufgaben 2., 3. und 4. können Sie auch das Programm *search.jar* verwenden. <http://aispace.org/search/version4.4.7/search.jar> )

1. Nehmen Sie an, dass zum Auffinden des Pfads „depth-first“-Suche mit Test auf Zyklen benutzt wird und dass die Auswahl der ausführbaren Aktionen in der Reihenfolge „nach oben“, „nach links“, „nach rechts“ und „nach unten“ erfolgt. Ein Knoten gelte als besucht, wenn er aus der „frontier“ genommen wird. (1 Pkt.)  
Nummerieren Sie die besuchten Knoten in der Reihenfolge Ihrer Abarbeitung. (Beginnen sie bei 1 und zählen hoch.)
2. Wieder ist ein Pfad von s nach g gesucht. Diesmal werde jedoch „best-first“-Suche benutzt. Man verwende die so genannte „Manhattan-Distanz“ zwischen zwei Punkten, die sich als Summe aus der Distanz in der x-Richtung und der Distanz in der y-Richtung ergibt. Es werde „multiple-path pruning“ vorausgesetzt.  
Nummerieren Sie die besuchten Knoten in der Reihenfolge Ihrer Abarbeitung. Welcher Pfad wird als erstes gefunden? (1 Pkt.)
3. Nun werde heuristische Tiefen-Suche mit Test auf Zyklen und der Manhattan-Distanz benutzt. Nummerieren Sie die besuchten Knoten in der Reihenfolge Ihrer Abarbeitung. Welcher Pfad wird gefunden? (1 Pkt.)

4. Für dieselbe Aufgabe werde der  $A^*$ -Algorithmus verwendet. Nummerieren Sie die besuchten Knoten in der Reihenfolge Ihrer Abarbeitung (bis maximal zum 20. Schritt). Welcher Pfad wird gefunden? (1 Pkt.)
5. Diskutieren Sie auf der Grundlage dieser Aufgaben, welcher der Suchalgorithmen für diese Aufgabenstellung geeignet ist. (2 Pkt.)
6. Nehmen Sie eine modifizierte Umgebung an, für die die relativen Positionen der verbotenen Felder und der Felder  $s$  und  $g$  gleich sind, für die jedoch keine (äußere) Begrenzung existiert. Welche der Methoden würde keinen Pfad mehr finden? Welche Methode ist für diese Umgebung geeignet? (2 Pkt.)

### Übungsaufgabe 3.2: (Regelbasierte Systeme: Türme von Hanoi)

Für die Darstellung der in einem regelbasierten System vorgegebenen und erschlossenen Fakten (kurz für die Elemente des Arbeitsspeichers) wird folgendes Format festgelegt:

(`typ attribut1 : wert1 ... attributN : wertN` )

Die Produktionsregeln haben die Form

`IF cond1 ... condN THEN aktion1 ... aktionM`

Das Antezedens besteht aus einer Liste von Bedingungen, das Sukzedens (oder Konsequenz) aus einer Liste von Aktionen. Eine Bedingung hat die Form

(`typ attribut1 : spez1 .... attributN : spezN`) bzw.

- (`typ attribut1 : spez1 .... attributN : spezN`).

Dabei ist - die Negation und jede Spezifikation ist entweder

- ein Atom,
- eine Variable,
- ein (arithmetisch) auswertbarer Ausdruck gekennzeichnet durch „[ ]“,
- ein boolescher Test gekennzeichnet mit „{ }“ oder
- die Konjunktion, Disjunktion oder Negation einer Spezifikation.

Beispielweise ist

(`person age: [n + 4] occupation: x`)

eine Bedingung, die genau dann durch ein Faktum des Typs `person` erfüllt wird, wenn es das Alter  $n + 4$  hat, wobei  $n$  an anderer bestimmt wurde.

Die Bedingung

- (`person age: {< 23 ^ > 6}`)

ist erfüllt, wenn es keine Person gibt, die älter als 6 und jünger als 26 ist.

Eine Aktion hat eine der folgenden Formen:

VON
6

- **ADD cond**: Das bedeutet, dass ein Faktenelement der Form **cond** direkt in die Faktenbasis eingefügt wird.
- **REMOVE i**: Entferne das Faktum, welches mit der *i*-ten Bedingung im Antezedent der Regel gematched wurde. (Nur anwendbar, wenn die Bedingung positiv ist).
- **MODIFY i (attribut spez)**: Modifiziere das Faktum, welche mit der *i*-ten Bedingung in der Regel gematched wurde, derart, dass der aktuelle Wert des Attributs **attribut** durch **spez** ersetzt wird. (Nur anwendbar auf positive Bedingungen. )

Nun zur eigentlichen Aufgabe, dem Hanoi-Problem:

Gegeben sind drei Pflocke A,B,C. Zu Anfang sind  $n$  Scheiben auf A angeordnet, so dass die kleinste Scheibe oben ist. Ziel ist es, die Scheiben in der richtigen Anordnung (mit kleinster Scheibe zuoberst) auf Pflock C zu übertragen. In jedem Schritt darf nur die oberste Scheibe bewegt werden; eine Scheibe kann nur von einem Pflock auf den anderen bewegt werden; und zu keinem Zeitpunkt darf eine größere Scheibe auf einer kleineren Scheibe sein.

Zur Lösung soll folgendes iterative Verfahren verwendet werden: Zuerst werden die Pflocke in einem Kreis angeordnet, so dass sich im Uhrzeigersinn die Folge A, B, C und dann wieder A ergibt. Es wird angenommen, dass keine Scheibe mehr als einmal nacheinander bewegt wird. Dann kann immer nur genau eine Scheibe ohne Verletzung der obigen Bedingungen bewegt werden. Die Scheibe wird auf den nächstmöglichen Pflock bewegt. Falls  $n$  gerade ist, geschieht die Suche nach dem nächstmöglichen Pflock im Uhrzeigersinn, falls  $n$  ungerade ist, gegen den Uhrzeigersinn.

Schreiben Sie eine Folge von Produktionsregeln, die dieses Verfahren implementiert und testen sie ihre Regeln anhand des Falls von drei Scheiben.

Zu Anfang sollte die Faktenbasis für alle  $1 \leq i \leq n$  Fakten der Form (**pflock: A scheibe: i**) und ein Faktum (**start**) enthalten. Falls keine Regel mehr feuern kann, sollte die Faktenbasis (**ende**) und für alle  $1 \leq i \leq n$  (**pflock: C scheibe: i**) enthalten.

*Version: 7. November 2008*

*Summe der erreichbaren Punkte auf diesem Blatt: 14*

*Summe der bisher erreichbaren Punkte: 42*