Grundlagen der Künstlichen Intelligenz

Prof. Dr. B. Nebel, Prof. Dr. W. Burgard
B. Frank, A. Karwath, G. Röger
Sommersemester 2009

Universität Freiburg
Institut für Informatik

Übungsblatt 3

Abgabe: Dienstag, 19. Mai 2009

Aufgabe 3.1 (Pfadplanung)

Betrachten Sie das Problem, den kürzesten Pfad zwischen zwei Punkten in einer Ebene zu finden, die konvexe Polygone als Hindernisse hat (siehe Abb. 1). Dies ist eine Idealisierung des Pfadplanungsproblems, das ein Roboter lösen muss, um in einer beengten Umgebung navigieren zu können.

(a) Angenommen, der Zustandsraum besteht aus allen Positionen (x, y) in der Ebene. Wieviele Zustaende gibt es? Wieviele mögliche Pfade zum Ziel gibt es?

(b) Erklären Sie kurz, warum der kürzeste Pfad von einer Polygonecke zu einer beliebigen anderen in der Umgebung aus (a) Liniensegmenten besteht, die (b) Ecken der Polygone verbinden. Definieren Sie nun einen geeigneten Zustandsraum. Wie groß ist dieser Zustandsraum?

(c) Um dieses Suchproblem zu implementieren, definieren Sie (in Text oder Pseudo Code) eine Nachfolger-Funktion, welche eine Ecke als Input bekommt und eine Menge von Ecken zurückliefert, die durch ein Liniensegment erreicht werden können.

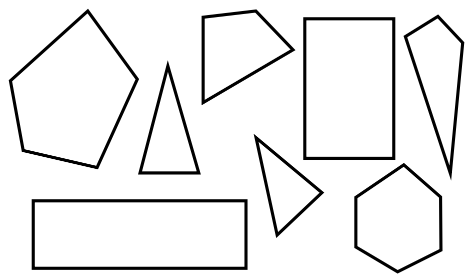

Abbildung 1: Roboternavigation unter Polygonen

Aufgabe 3.2 (Lokale Suche)

Wir werden nun *Hill-Climbing* im selben Zusammenhang (Roboternavigation in der Ebene unter polygonalen Hindernissen) untersuchen.

(a) Erklären Sie, wie *hill-climbing* als Methode zum Erreichen eines bestimmten Punktes in der Ebene durchgeführt werden würde.

(b) Erläutern Sie mit Hilfe eines Beispiels, warum nicht-konvexe Hindernisse zu einem lokalen Maximum für den Algorithmus führen kann.

(c) Ist es möglich, bei konvexen Hindernissen zu einer nicht-optimalen Lösung zu gelangen?

(d) Würde *simulated annealing* bei dieser Problemfamilie immer aus lokalen Maxima herausfinden? Begründen Sie Ihre Antwort.

Aufgabe 3.3 (CSPs)

Das $SEND+MORE = MONEY$-Problem besteht darin, paarweise verschiedene Ziffern für die Buchstaben D, E, M, N, O, R, S, Y zu finden, so dass S und M ungleich Null sind, d.h. keine führenden Nullen, und dass die Gleichung

$$SEND + MORE = MONEY$$

erfüllt ist.

(a) Erklären Sie, warum es sinnvoll ist, dieses Problem als *constraint satisfaction problem* zu formulieren!

(b) Formulieren Sie das Problem als *constraint satisfaction problem*, d.h. geben Sie die Variablen, Constraints, etc. an.

(c) Suchen Sie unter Benutzung der in der Vorlesung vorgestellten *forward checking* und *arc consistency* Methoden nach einer Lösung und geben Sie den Suchbaum an.
(Hinweis: Betrachten Sie die Buchstaben in der folgenden Reihenfolge: O, M, Y, E, N, D, R, S.)

Aufgabe 3.4 (Kantenkonsistenz)

AC-3 fügt *jede* Kante (X_k, X_i) wieder in die Warteschlange ein, wenn *irgendein* Wert von X_i gelöscht wird, selbst wenn jeder Wert von X_k noch mit mehreren verbleibenden Werten von X_i konsistent ist. Angenommen, wir führen für jede Kante (X_k, X_i) und jeden Wert von X_k darüber Buch, wieviele verbleibende Werte von X_i mit X_k konsistent sind. Erklären Sie, wie diese Zahlen effizient aktualisiert werden können und zeigen Sie damit, dass Kantenkonsistenz in Zeit $O(n^2d^2)$ garantiert werden kann.

Die Übungsblätter dürfen und sollten in Gruppen von drei (3) Studenten bearbeitet werden. Bitte füllen Sie das Deckblatt[1] aus und heften Sie es an Ihre Lösung.

[1] http://www.informatik.uni-freiburg.de/~ki/teaching/ss09/gki/coverSheet-german.pdf