

GWV-Abgabe zum 17.10.2014

Arne Struck, Knut Götz

24. Oktober 2014

1.1

1.

In dem Fall einer Anwendung wäre der State-Space eines zu erstellenden Graphens, die Menge aller Haltestellen (Busse, Bahnen etc.) des öffentlichen Nahverkehrs. Jeder Knoten repräsentiert eine Haltestellen und jede Kante zwischen zwei Knoten repräsentiert eine Verbindung zwischen zwei Haltestellen. Gibt es beispielsweise zwei Haltestellen A und B und eine Buslinie verkehrt zwischen diesen müsste dies im Graphen durch eine Kante repräsentiert werden.

2.

a)

siehe java-files.

Das Programm im Rohzustand erwartet keine Nutzereingaben, es gibt einen der kürzesten Wege zum gewünschten Zustand aus. Sollte man allerdings das Verlangen spüren mit ihm "zu spielen" bieten die Dateien node.java für die Bechergößen und start.java für die Zielmenge im (gedacht) größeren Behälter in Lieter, ob verschütten möglich sein soll und ein wenig unterschiedlichen output Optionen in Form von Konstanten.

b)

Educated Guess:

Ohne Wegschütten ist dies nicht mehr möglich, da zu viele Übergänge zwischen den verschiedenen Stati wegfallen. Es existiert eine Möglichkeit 2 Lieter in den 4 Lieter Behälter zu gießen ohne Flüssigkeit zu verschwenden, aber auch diese kommt nicht ohne Wegschütten des Restes aus, wenn die 2 Lieter aus dem 3 Lieter Behälter transferiert werden soll.

Programm mit entsprechender Option:

no connection found between source and target => impossible to solve

1.2

Eines unserer Beispiele für künstliche Intelligenz bei dem mit Sicherheit eine Art von Suche vorkommt ist der Schachcomputer. Der Suchraum ist in diesem Fall die Menge aller möglichen Stellungen (eine Stellungen beschreibt eine zulässige Positionierung von Spielfiguren auf dem Schachbrett und welche Partei am Zug ist), die sich aus der aktuellen Stellung, der Stellung in der die Suche gestartet wird, ergeben können. Bei einem neu beginnenden Spiel wäre die Startstellung beim

Schachspiel (alle Bauern auf der 2. bzw. 7. Reihe, Weiß am Zug etc.) und der die Stellungen repräsentierende Zustand, der Startzustand. Die Menge der Zielzustände entspricht dann der Menge der Stellungen bei denen der Schachcomputer seinen Gegner schachmatt setzen kann und das Spiel gewinnt. 0 Die Menge der Kanten entspricht, dann der Menge der möglichen Züge. Zwei Zustände, die jeweils eine Stellung repräsentieren, sind genau dann mit einer Kante im Suchgraphen verbunden, wenn es einen zulässigen Zug in der ersten Stellung gibt aus der sich die zweite Stellung ergibt. So würde sich dann ein Baum ergeben, dessen Wurzel der Startzustand ist und dessen Blätter alle Stellungen sind in der eine Partie schachmatt ist. Der Baum ist endlich, da es sich nicht unendlich viele mögliche Stellungen gibt.

Schwierigkeiten ergeben sich dadurch, dass zum einen der Schachcomputer, sollte er nicht gegen sich selbst spielen, nur eine Verzweigung der Suche steuern kann. Das größere Problem bei einer solchen Suche ist jedoch, dass es in der Regel nicht möglich ist, aus einer gegebenen Stellungen zuerst den kompletten Baum aufzubauen, da der Baum meist sehr groß ist. Es müsste also eine Heuristik geben mithilfe derer die einzelnen Zustände bewertet werden. Der Schachcomputer könnte sich mithilfe der Suche dann so entscheiden, dass er den Zug wählt, der in den besten Teilbaum führt. Der beste Teilbaum wird so bestimmt, dass er auf jeden Zug des Gegners reagieren kann, so dass ein Pfad gewählt werden kann dessen Endknoten von der Heuristik besser bewertet wird, als der Endknoten entsprechender Pfade in anderen Teilbäumen.