Tutorial 1: Search Space design

Finn-Lasse Jörgensen, Frederik Wille, Tronje Krabbe

October 24, 2016

Exercise 1.1 (Search space properties)

1. Fully observable & partially observable In einer partially observable Umgebung muss ein Agent stets Annahmen über den Zustand dieser machen, da er diesen ja nicht vollständig sehen kann; sind diese Annahmen falsch, kann es zu schlechten Entscheidungen und unerwünschten Ergebnissen kommen.

   In einer fully observable Umgebung hingegen ist der gesamte Zustand dem Agenten bekannt, jedoch kann ein Problem z.B. sein, dass die resultierende Datenmenge zu groß und somit nicht verarbeitbar ist.

   Ein Beispiel für ein partially observable environment wäre etwa ein Kartenspiel wie Skat, wo der Agent die eigenen sowie alle bisher abgelegten Karten kennt, nicht aber die der Mitspieler.

   Ein fully observable environment ist im Brettspiel Go zu finden, jedoch ist die Anzahl möglicher Züge und deren Auswirkungen, etc. so groß, dass ein AI wie AlphaGo enorme Rechenressourcen benötigt, um effektiv zu spielen.

2. Discrete & continuous Diskrete Daten nehmen immer bestimmte Werte an, wohingegen kontinuierliche Daten beliebige Werte annehmen können, und zwischen zwei Werten noch unendlich viele weitere Werte liegen. Ein Würfel beispielsweise produziert diskrete Daten, nämlich die Werte 1, 2, 3, 4, 5, oder 6. Misst man die Zeit, die für eine Aktion benötigt wird, produziert dies kontinuierliche Daten. Es gibt beliebig kleine Zeitschritte.

   Oft ist es sinnvoll, kontinuierliche Daten als diskrete zu behandeln; misst man tatsächlich Zeit, so ist es sinnvoll, eine gewisse Genauigkeit auszuwählen, und dann zu runden. Beschränkt man sich auf etwa Millisekunden, geht man nur noch mit diskreten Werten um.

Für ein Programm, wie etwa eine AI, muss man, arbeitet man in kontinuierlichen Umgebungen, Daten zunächst diskretisieren, damit zie benutzbar und verarbeitbar werden.

3. Deterministic & stochastic Ein Lichtschalter produziert ein deterministisches Ergebnis, das werfen eines Würfels ein stochastisches. Ähnlich wie bei fully und partialy observable Umgebungen, muss man in einem stochastischen System gewissen Annahmen machen, kann aber nicht sicher ein optimales Ergebnis erzielen. Ein deterministisches System ist einfach umzusetzen, jedoch, denkt man z.B. an Cryptographie, eventuell unsicher.

Exercise 1.2 (Search space 1)

1. Wenn man eine rein geographische Route planen möchte, wäre der Zustandsraum alle Haltestellen des Nahverkehrs. Dann sind die Knoten die Haltestellen, die Kanten sind alle Strecken zwischen zwei Stationen, die ohne Zwischenhalt gefahren werden und die Kantengewichte repräsentieren die Fahrtdauer zwischen den beiden.

2. a) Das Modell repräsentiert den Füllstand der beiden Jugs als Tuple mit dem 4-Liter Jug als erste und dem 3-Liter Jug als zweite Stelle. Die Werte des Tupels sind in Litern, somit ist 0 für beide Stellen der minimale und 4 bzw 3 (4,3) die maximalen Werte. Der Startzustand ist (0,0). Übergänge entstehen durch Füllen eines Jugs mit dem Zapfhahn, entleeren eines Jugs oder durch Umfüllen eines Jugs in den anderen, wobei entweder der befüllte Jug komplett gefüllt oder der füllende Jug komplett entleert werden muss. Mögliche Zustände sind: (0,0), (4,0), (1,3), (4,3), (0,3), (3,0), (3,3), (4,2), (0,2), (2,0). Das Ziel ist der Zustand (2,0). Die Zustandsreihenfolge um das Rätsel zu lösen ist: (0,0), (0,3), (3,0), (3,3), (4,2), (0,2), (2,0)

   b) Das Rätsel ist immer noch genauso lösbar, nur trinkt man den Wein anstatt ihn wegzukippen. Dass dies eventuell länger dauert, ist zu vernachlässigen.