

## 1 Ćwiczenie 2

1. Zapewne pamiętasz o odkurzaczu omówionym w pierwszych dwóch wykładach.  
(i) Sformułuj problem jako problem przeszukiwania; tzn. musisz napisać stan początkowy, możliwe działania za pomocą funkcji **Action(state)**, relację przejścia za pomocą funkcji **Result(state, action)** i stan celu. Proszę zobaczyć strona 9 wykładu 2. Zakładamy, że obie kwadraty A i B są brudne i agent znajduje się w kwadracie A. Zakładamy również, że agent ma trzy możliwe działania: ruszaj w lewo, ruszaj w prawo i ssać. (ii) Napisz program za pomocą algorytmu 'Breadth First Search, aby agent mógł odszukiwać brud we wszystkich możliwych lokalizacjach i wyczyścić go (tzn., znalezienie ścieżki z węzła początkowego do węzła docelowego. Wystarczy, aby program wydrukował każdą sekwencję akcji na konsoli, i pokazał kiedy osiąga cel (czyli ścieżka z węzła początkowego do węzła docelowego).

2. Rozważ 8 zagadek ze stanem początkowym jako  $\begin{bmatrix} 1 & 1 & 3 \\ 4 & 2 & 5 \\ 7 & 8 & 6 \end{bmatrix}$ .

Stan docelowy to:  $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & \cdot \end{bmatrix}$ .

- (i) Opisz sformułowanie problemu jako problem przeszukiwania.
  - (ii) Napisz program, używając przeszukiwania  $A^*$  (Zobacz numer wykładu 3). Uważamy, że każde kroki kosztuje 1, a zatem koszty osiągnięcia stanu  $n$  z stanu początkowego (czyli  $g(n)$ ) jest całkowita liczba kroków wymaganych do osiągnięcia stanu  $n$ . Możesz użyć funkcji heurystycznej, gdzie  $h(\text{stan}) = \text{suma odległości bloku miasta (sum of city block distance)}$ . Wystarczy, aby program wydrukował każdą sekwencję akcji na konsoli, i pokazał kiedy osiąga cel.  
(Uwaga: W celu wykonania (ii) byłoby dobrze stworzyć drzewo gry z powiązaną wartością  $f(n) = g(n) + h(n)$  dla każdego węzła.)
3. Zapewne pamiętasz o Trywialnej grze rozważanej w wykładach 3 i 4. Napisz program, aby znaleźć optymalną strategię dla MAX za pomocą algorytmu Maximin z przycinaniem  $[\alpha, \beta]$  ( $[\alpha, \beta]$ -pruning).