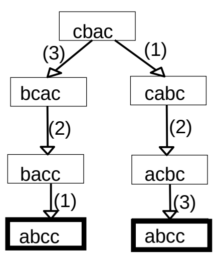
Wnioskowanie
jako
przeszukiwanie przestrzeni stanów

Plan wykładu

- Rozwiązywanie problemów jako poszukiwanie ścieżki rozwiązania
- Przestrzeń stanów jako graf skierowany
- Dokładne metody przeszukiwania przestrzeni stanów
- Przybliżone metody przeszukiwania przestrzeni stanów

Systemy produkcyjne

ba→ab	(1)
ca→ac	(2)
cb→bc	(3)



Przestrzeń stanów jest to czwórka uporządkowana **[N, A, S, GD],** gdzie:

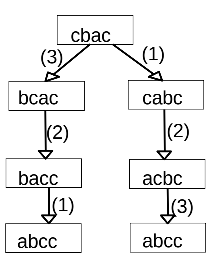
N jest zbiorem wierzchołków odpowiadających stanom w procesie rozwiązywania problemu

A jest zbiorem krawędzi, odpowiadających krokom w procesie rozwiązywania problemu

S jest niepustym podzbiorem N, zawierającym stany początkowe problemu

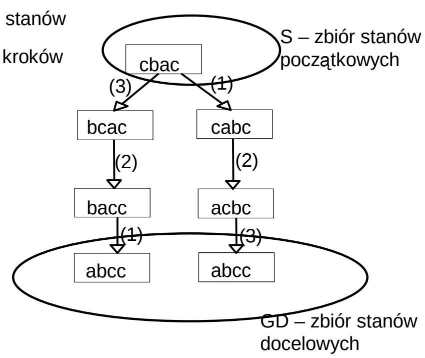
GD jest niepustym podzbiorem N, zawierającym stany docelowe problemu.

N – zbiór stanów



N – zbiór stanów

A – zbiór kroków



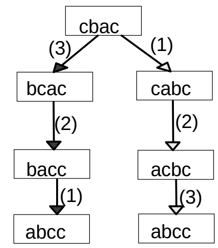
Sztuczna inteligencja

Stany w GD są opisane:

1. przez podanie własności stanów występujących w przeszukiwaniu
2. przez podanie własności ścieżki tworzonej podczas przeszukiwania

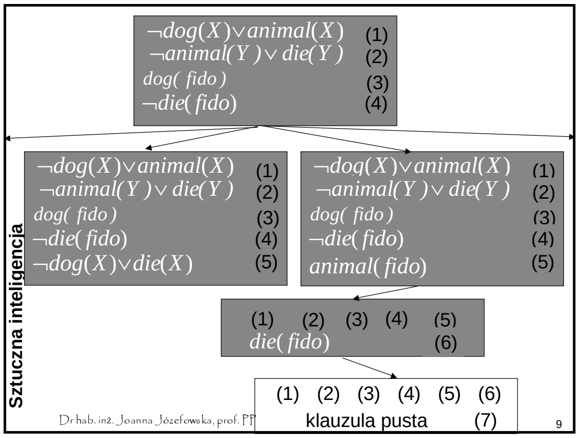
Ścieżką rozwiązania nazywamy ścieżkę wiodącą przez ten graf z wierzchołka z S do wierzchołka w GD.

Sztuczna inteligencja



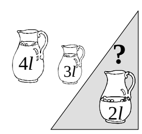
Dwie ścieżki rozwiązania

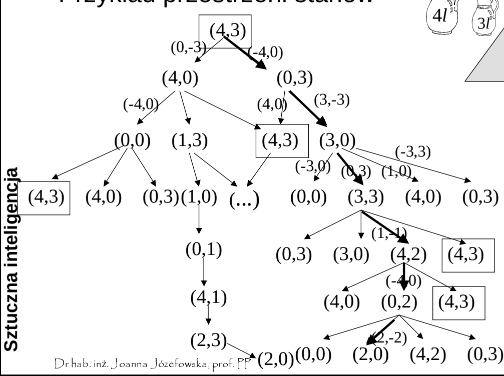
Sztuczna inteligencja



Sztuczna inteligencja

Przykład przestrzeni stanów





Sztuczna inteligencja

Metody rozwiązywania problemów

- Czy metoda gwarantuje znalezienie rozwiązania?
- Czy algorytm zakończy się w każdym przypadku, czy może wpaść w pętlę nieskończoną?
- Czy jeśli rozwiązanie zostanie znalezione, to mamy gwarancję, że jest ono optymalne?
- Jaka jest czasowa i pamięciowa złożoność obliczeniowa algorytmu?
- W jaki sposób interpreter może najlepiej zredukować złożoność przeszukiwania?
- Jak można zaprojektować interpreter najbardziej efektywny z punktu widzenia wykorzystanego języka reprezentacji wiedzy?

Sztuczna inteligencja

Algorytmy przeszukiwania

Zadaniem algorytmów przeszukiwania jest znalezienie ścieżki rozwiązania w przestrzeni problemu.

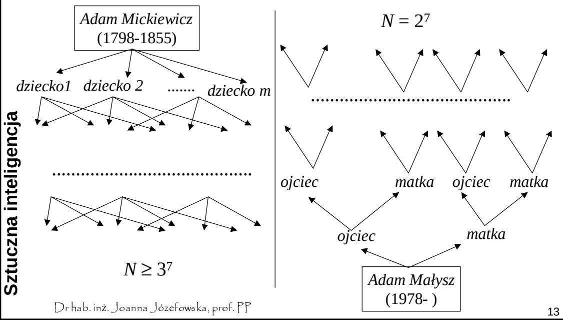
Strategie przeszukiwania:

- wszerz
- w głąb
- w przód
- w tył

Implementacja przeszukiwania: rekurencja.

Strategia przeszukiwania a złożoność

Adam Małysz jest potomkiem Adama Mickiewicza.



Kiedy stosować przeszukiwanie w tył?

- Gdy cel lub hipoteza jest dana w sformułowaniu problemu, np. w dowodzeniu twierdzeń matematycznych, systemach diagnostycznych.
- Gdy liczba reguł możliwych do zastosowania rośnie szybko, a wczesna eliminacja celów może wyeliminować przeszukiwanie pewnych gałęzi.
- Gdy stan początkowy nie jest dany explicite, ale musi być rozpoznany. Przeszukiwanie wstecz może pomóc w pokierowaniu pozyskiwaniem danych (np. diagnostyka medyczna).

Kiedy stosować przeszukiwanie w przód?

- Gdy wszystkie lub większość danych jest zawartych w sformułowaniu problemu (np. interpretacja).
- Gdy występuje duża liczba potencjalnych celów, ale jest tylko kilka możliwości zastosowania faktów i informacji wejściowych do konkretnej instancji problemu.
- Gdy trudno jest sformułować hipotezę docelową.

Przeszukiwanie heurystyczne

Funkcja oceny heurystycznej

$$f(n) = g(n) + h(n)$$

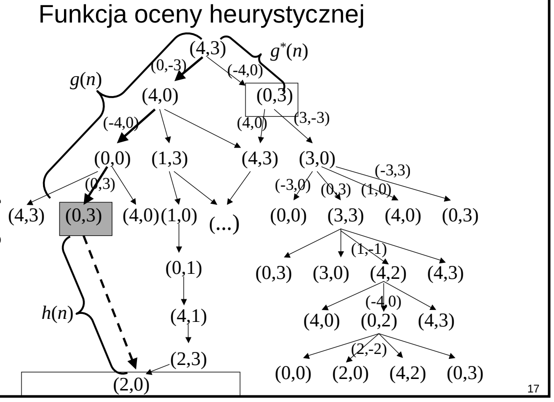
gdzie:

$g(n)$ mierzy aktualną długość ścieżki od stanu n do stanu początkowego

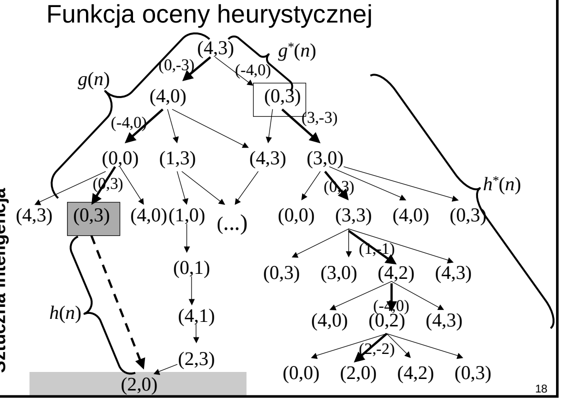
$h(n)$ jest heurystycznym oszacowaniem odległości stanu n od celu.

Jeżeli taka funkcja jest realizowana ze strategią przeszukiwania *best-first-search*, to algorytm nazywa się algorytmem A.

Funkcja oceny heurystycznej



Funkcja oceny heurystycznej



Sztuczna inteligencja

Algorytm A*

Jeżeli algorytm A wykorzystuje funkcję oceny heurystycznej taka, że $h(n) \le h^*(n)$, to otrzymany w ten sposób algorytm nazywa się algorytmem A*.

Sztuczna inteligencja

Dopuszczalność

Algorytm przeszukiwania jest dopuszczalny (*admissible*), gdy gwarantuje znalezienie najkrótszej ścieżki do rozwiązania, jeżeli taka ścieżka istnieje.

Wszystkie algorytmy A* są dopuszczalne.

Algorytm przeszukiwania wszerz jest algorytmem A*.

$$f(n) = g(n) + 0$$

Sztuczna inteligencja

Monotoniczność

Funkcja $h(n)$ jest monotoniczna, gdy:

1. $h(n_i) - h(n_j) \le \mathrm{cost}(n_i, n_j)$

gdzie $\mathrm{cost}(n_i, n_j)$ jest rzeczywistym kosztem (liczbą kroków) przejśćia od stanu n_i do stanu n_j

2. Ocena heurystyczna stanu docelowego $h(goal) = 0$.

Heurystyka monotoniczna osiąga każdy stan najkrótszą ścieżką.

Sztuczna inteligencja

Twierdzenie 1

Każda heurystyka monotoniczna jest dopuszczalna.

DOWÓD

Niech $s_1, s_2, \ldots, s_g$ będzie dowolną ścieżką od jakiegoś stanu początkowego s_1 do celu s_g.

Na podstawie definicji monotoniczności, dla każdej pary kolejnych stanów na tej ścieżce zachodzi:

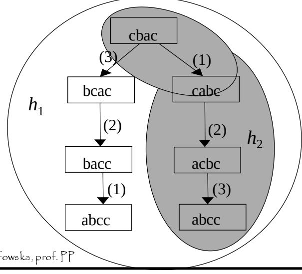
$$h(s_1) - h(s_2) \le \mathrm{cost}(s_1, s_2)$$
$$h(s_2) - h(s_3) \le \mathrm{cost}(s_2, s_3)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$h(s_{g-1}) - h(s_g) \le \mathrm{cost}(s_{g-1}, s_g)$$
$$h(s_1) - h(s_g) \le \mathrm{cost}(s_1, s_g) = h^*(s_1) \qquad \boxed{h(s_g) = 0}$$

$$h(s_1) \le h^*(s_1)$$

Sztuczna inteligencja

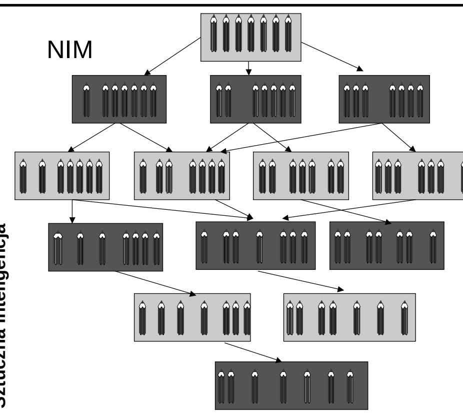
Poinformowanie

Dla dwóch heurystyk h_1 i h_2 typu A*, jeżeli $h_1(n) \le h_2(n)$ dla wszystkich stanów w przeszukiwanej przestrzeni, to o h_2 mówimy, że jest lepiej poinformowana niż h_1.



Sztuczna inteligencja

NIM



Procedura MINiMAX

Przeciwnicy posiadają taką samą wiedzę o przestrzeni stanów i stosują tę wiedzę konsekwentnie w celu wygrania gry.

Jeżeli ojcem jest MIN, to nadaj mu najmniejszą wartość spośród dzieci.

Jeżeli ojcem jest MAX, to nadaj mu największą wartość spośród dzieci.

Podsumowanie

- Przeszukiwanie przestrzeni stanów jest ogólnym modelem rozwiązywania problemów
- Algorytmy dokładne przeszukiwania są zwykle zbyt pracochłonne
- Algorytmy heurystyczne wymagają dostosowania do charakterystyki rozwiązywanego problemu
- Można zdefiniować ogólne własności heurystyk: dopuszczalność, monotoniczność i informatywność