

ALGORYTM A*

---podstawowe wiadomości - przypomnienie

1. Opis problemu w kategoriach pojęciowych przestrzeni stanów

Koncepcja **przeszukiwania przestrzeni stanów** (*state space search*) jako metody rozwiązywania problemów polega na reprezentowaniu dziedziny problemu jako przestrzeni stanów, w której będzie prowadzone poszukiwanie stanu reprezentującego rozwiązanie.

Zdefiniowanie przestrzeni stanów dla dziedziny problemu wymaga

- określenia **postaci opisu stanu** w dziedzinie problemu
- podania **stanu początkowego**, od którego rozpoczyna się poszukiwanie rozwiązania
- zdefiniowania funkcji **następnik** (*successor function*), która każdemu stanowi w dziedzinie problemu przypisuje zbiór par $\langle \text{możliwa akcja}, \text{następny stan} \rangle$
 - a więc dla każdego stanu określa akcje możliwe w tym stanie, a dla każdej akcji nowy stan osiągnięty w wyniku jej wykonania
- zdefiniowania funkcji **koszt kroku** (*step cost*), która każdemu stanowi przypisuje koszt przejścia do tego stanu ze stanu rodzica
- zdefiniowania **kryterium celu** (*goal condition*), czyli warunku/warunków, jaki/jakie musi spełniać stan, aby został rozpoznany jako **cel przeszukiwania** przestrzeni stanów, tzn. jako stan reprezentujący rozwiązanie

W przypadku zastosowania strategii przeszukiwania z heurystyczną oceną stanów, opis przestrzeni stanów wymaga ponadto zdefiniowania funkcji przypisującej ocenę każdej ścieżce w przestrzeni stanów.

2. Przeszukiwanie przestrzeni stanów

Przeszukiwanie przestrzeni stanów reprezentującej dziedzinę problemu polega na rozpatrywaniu kolejnych stanów tej przestrzeni -począwszy od stanu początkowego- aż do znalezienia stanu spełniającego kryterium celu. Kolejne stany do rozpatrzenia są osiągnięte przez powtarzane użycie funkcji **następnik**: w każdym kroku przeszukiwania wynikiem zastosowania funkcji następnik do stanu rozpatrywanego w tym kroku są kolejne stany, które zostają dodane do zbioru stanów do rozpatrzenia.

3. Wybór stanu do rozpatrzenia

W każdym kroku przeszukiwania przestrzeni stanów spośród wszystkich dotychczas osiągniętych stanów w ogólności tylko część stanów została już rozpatrzona, dlatego że każdy stan może mieć wiele stanów potomnych, a rozpatrywany w każdym kroku jest tylko jeden stan. Wybór stanu do rozpatrzenia spośród wszystkich dotychczas osiągniętych zależy od **strategii przeszukiwania**.

W przypadku strategii **przeszukiwania wszerz** (*breadth first*) oraz w przypadku strategii **przeszukiwania z heurystyczną oceną stanów** (*best first*) wszystkie stany osiągnięte od początku przeszukiwania, a jeszcze nie rozpatrzone, są ustawiane w **kolejkę**. W przypadku strategii **wszerz** jest to zwykła kolejka LIFO, natomiast w przypadku strategii **best first** stany w kolejce są uszeregowane według oceny heurystycznej – jest to tzw. **kolejka priorytetowa** (*priority queue*). W każdym kroku do rozpatrzenia jest pobierany stan z czoła kolejki.

4. Drzewo poszukiwań

Algorytmy przeszukiwania typu **wszerz** (*breadth first*) oraz **best first** są typowo implementowane jako algorytmy konstruuujące **drzewo poszukiwań** (*search tree*). Każdy węzeł drzewa poszukiwań (*search tree node*) reprezentuje osiągnięcie pewnego stanu przestrzeni stanów **określoną ścieżką od węzła początkowego**. Atrybutami węzła drzewa poszukiwań są

- stan przestrzeni stanów, którego osiągnięcie reprezentuje dany węzeł
- stan rodzic, tzn. stan, którego potomkiem jest stan związany z danym węzłem
- koszt osiągnięcia stanu związanego z danym węzłem od stanu początkowego, ścieżką reprezentowaną przez ten węzeł
- akcja powodująca przejście do stanu związanego z danym węzłem ze stanu rodzica

Dodatkowy atrybut węzła w przypadku algorytmu typu **best first** :

- ocena heurystyczna stanu osiągniętego ścieżką reprezentowaną przez ten węzeł

5. Osiąganie stanu różnymi ścieżkami od stanu początkowego reprezentowane przez różne węzły drzewa poszukiwań

W przestrzeni stanów w ogólności każdy stan może być osiągnięty ze stanu początkowego różnymi drogami, ponieważ każdy stan może mieć więcej niż jednego rodzica. Przejścia między stanami tworzą więc w ogólności graf, który niekoniecznie jest drzewem. W konsekwencji w **drzewie poszukiwań** jeden stan przestrzeni stanów może być reprezentowany przez więcej niż jeden węzeł – różne węzły drzewa poszukiwań związane z tym samym stanem reprezentują różne ścieżki osiągnięcia tego stanu od stanu początkowego.

Skoro różne ścieżki osiągnięcia stanu są reprezentowane przez różne węzły w przestrzeni poszukiwań, to przejścia między węzłami zawsze tworzą drzewo –stad termin **drzewo poszukiwań**–niezależnie od tego, czy struktura utworzona przez połączone stany w przestrzeni stanów jest drzewem, czy też nie.

6. Algorytm A*

Algorytm A* jest algorytmem przeszukiwania przestrzeni stanów typu **best first** z zastosowaniem oceny heurystycznej postaci

$$f(Węzeł) = g(Węzeł) + h(Węzeł)$$

gdzie

$g(Węzeł)$ jest to koszt dojścia do stanu związanego z węzłem drzewa poszukiwań od stanu początkowego

$h(Węzeł)$ jest to szacowany koszt dojścia z tego stanu do stanu spełniającego kryterium celu

Ocena jest przypisana do węzła drzewa poszukiwań, ponieważ składowa **g** zależy w ogólności od ścieżki dojścia ze stanu początkowego, a więc nie jest jednoznaczna funkcją stanu.

Jako wymaganie dotyczące oceny heurystycznej przyjmuje się, że funkcja **f** musi być oceną **”nie pesymistyczną”**, tzn. nie przeceniającą rzeczywistego całkowitego kosztu dojścia ze stanu początkowego do stanu docelowego przez stan związany z ocenianym węzłem (oczywiście przyjmując ścieżkę od stanu początkowego reprezentowaną przez ten węzeł). Jeśli funkcja **f** spełnia ten warunek, to -jak łatwo wykazać- jeśli rozwiązanie istnieje, algorytm zawsze poda rozwiązanie o najniższym koszcie.

7. Problem powtórzeń stanów : wybór między węzłami związanymi z tym samym stanem

Jeśli algorytm w trakcie poszukiwania osiągnie stan przestrzeni stanów, który został wcześniej osiągnięty inną ścieżką, to dalsze budowanie drzewa poszukiwań od otrzymanego węzła może być źródłem nieefektywności, ponieważ poddrzewo zakorzenione w tym węźle mogłoby być powtórzeniem poddrzewa zbudowanego wcześniej. Rozwiązanie tego problemu najczęściej jest dość proste: można wykazać, że jeśli funkcja **f** jest **monotoniczna**, to z dwóch węzłów drzewa poszukiwań związanych z tym samym stanem węzeł osiągnięty później zawsze reprezentuje bardziej kosztowną ścieżkę. A więc węzeł ze stanem **powtórzonym** – stanem związanym z otrzymanym wcześniej węzłem – można odrzucić, tzn. nie wstawiać go do kolejki węzłów do rozpatrzenia.

Implementacje algorytmu A* często przyjmują takie rozwiązanie problemu powtórzeń stanów, ponieważ w praktyce okazuje się, że ocena **f** która nie jest pesymistyczna, najczęściej jest też monotoniczna.