Програмиране в изкуствения интелект—Зимен семестър 1999/2000.
Преподавател: Иван А. Держански.

Лекция 3: Алгоритми за търсене в пространство от състояния

1 Алгоритми за търсене

- Общ алгоритъм — реализация чрез използуване на някаква структура от данни, в която се съхранява **фронтът** на дървото на търсене (множеството от активни състояния — тези, които още не са развити).
- Параметри на алгоритмите: **пълнота**, **оптималност** и **сложност** (**по време** = брой изследвани възли и **по памет** = максимален размер на фронта).

  Избягване на повторения и зацикляне.
- Параметри на дървото на търсене: **дълбочина** d, **разклоненост** b.

2 Алгоритми за пълно изчерпване

- Търсене в ширина (breadth-first) — използува се опашка. Стратегията е пълна и оптимална, но сложността по памет е $O(b^d)$. Сложността по време е също $O(b^d)$, защото в най-неблагоприятния случай може да се наложи да се претърси цялото дърво от $\dfrac{b^{d+1}-1}{b-1}$ възела.

  Вариант: търсене с равномерна цена на пътя (uniform-cost search) — сортиране на множеството частични траектории по сумарната цена на изминатия път вместо по броя ходове.
- Търсене в дълбочина (depth-first) — използува се стек. Сложността по памет е $O(bd)$, но стратегията не е нито пълна, нито оптимална.

  Вариант: итеративно търсене по нива (iterative deepening) — по същество търсене в дълбочина във все по-дълбока част от дървото. Стратегията е пълна и оптимална и разходът на памет е $O(bd)$, но търсенето изисква повече време — преглеждат се $\sum_{j=0}^{d} \dfrac{b^{j+1}-1}{b-1}$ възела, т. е. $\dfrac{b}{b-1}$ пъти повече, отколкото има в дървото.

3 Обща структура на търсещата програма на Пролог

```
решение( S, []) :- цел( S).
решение( S, [Head | Tail]) :- ход( NewS, Head), решение( S, Tail).
```

Задачата се решава с извикване на `нач( S), решение( S, Path)`. Дефинициите на `нач/1`, `ход/2` и `цел/1` зависят от задачата.