Севастопольский государственный университет
Институт информационных технологий

“МЕТОДЫ И СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА"
(МиСИИ)

Бондарев Владимир Николаевич

Лекция 3

Агенты, решающие задачи ИИ. Представление задач в пространстве состояний.

В.Бондарев

Способы представления задач

Термин **"решение задач"** (problem solving) употребляется в искусственном интеллекте в весьма ограниченном смысле. Речь идет о хорошо определенных задачах, решаемых на основе поисковых алгоритмов.

Задача считается хорошо определенной, если для неё имеется возможность задать пространство возможных решений (состояний), а также способ просмотра этого пространства с целью поиска конечного (целевого) состояния, соответствующего решенной задаче.

Рассматриваются **способы представления задач**, удобные для их решения на ЭВМ. К ним относятся следующие наиболее часто используемые способы:

- представление задач в пространстве состояний;

- представление, сводящее задачу к подзадачам;

- представление задач в виде теорем.

В.Бондарев

Представление задач в пространстве состояний

Подход, использующий пространство состояний, весьма распространен в решении задач. Он включает в себя:

- задание начальных состояний задачи;
- задание конечных (целевых) состояний задачи;
- задание операторов, преобразующих одни состояния задачи в другие.

Решение задачи состоит в том, чтобы найти последовательность операторов, преобразующих начальное состояние задачи в конечное, которое соответствует решенной задаче.

Существуют многочисленные задачи, решение которых интерпретируется как поиск в пространстве состояний: поиск пути на карте; задача преобразования сцен; выделение границ на изображении; различные игры и головоломки.

Задача поиска в пространстве состояний

В общем случае задача, поставленная как задача поиска в пространстве состояний, определяется совокупностью четырех составляющих

$$(S0, S, F, G), \quad (1)$$

где S – множество возможных состояний задачи;

$S0$ – множество начальных состояний, $S0 \in S$;

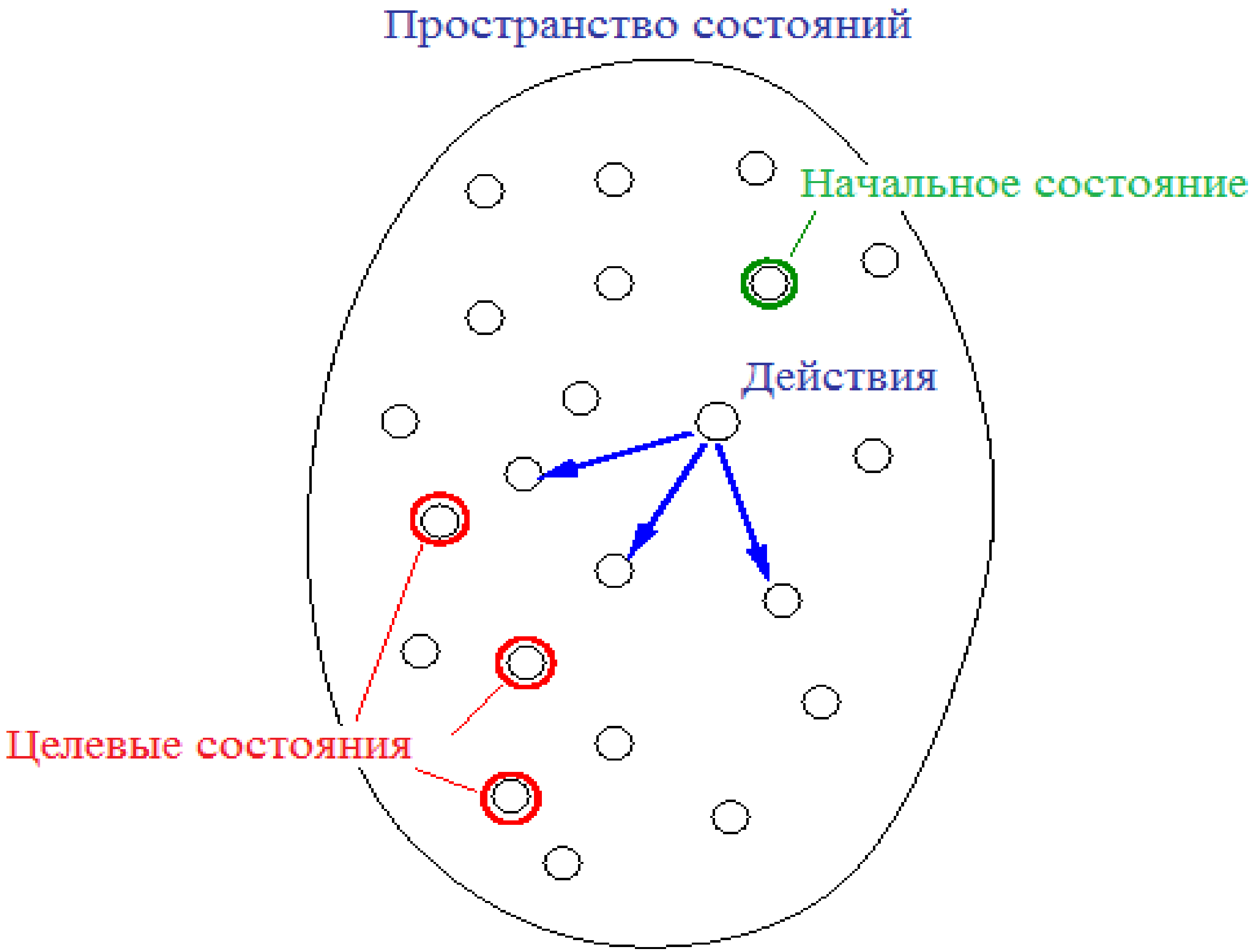
F – множество операторов, преобразующих состояния ;

G – множество целевых состояний, $G \in S$.

Каждый **оператор** $f \in F$ является функцией, отображающей одно состояние в другое – $s_j = f(s_i)$, где $s_i, s_j \in S$. **Решением задачи** является последовательность операторов $f_i \in F$, преобразующих начальные состояния в конечные, т.е.

$$f_n(f_{n-1}(\ldots(f_2(f_1(S0)))\ldots)) \in G .$$

Задача поиска в пространстве состояний



Необходимо найти последовательность действий, преобразующих начальное состояние в целевое, которую называют **планом**.

Поиск в пространстве состояний

Состояния задачи можно представлять в виде **графа состояний**. Множество вершин графа соответствует состояниям задачи, а множество дуг (ребер) – операторам. В этом случае поиск решения задачи может интерпретироваться как поиск пути на графе.

Поиск в пространстве состояний

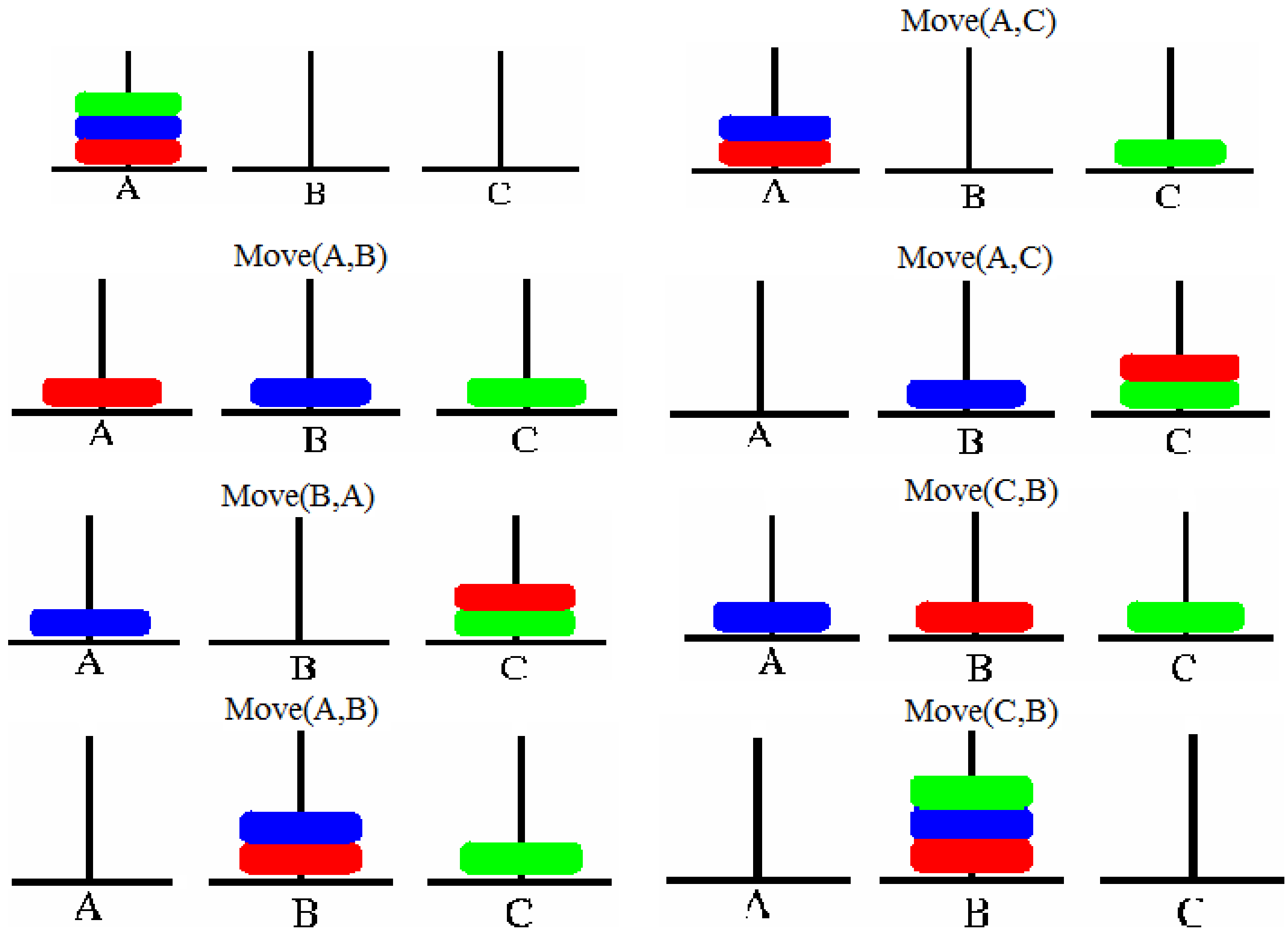
Обобщенный процесс поиска решения:

1. Принять начальное состояние за текущее;
2. Проверить, не соответствует ли текущее состояние целевому;
3. Выполнить допустимые операторы и найти новые состояния-приемники;
4. Выбрать одно из новых состояний-приемников и принять его за текущее состояние и перейти к пункту 2.

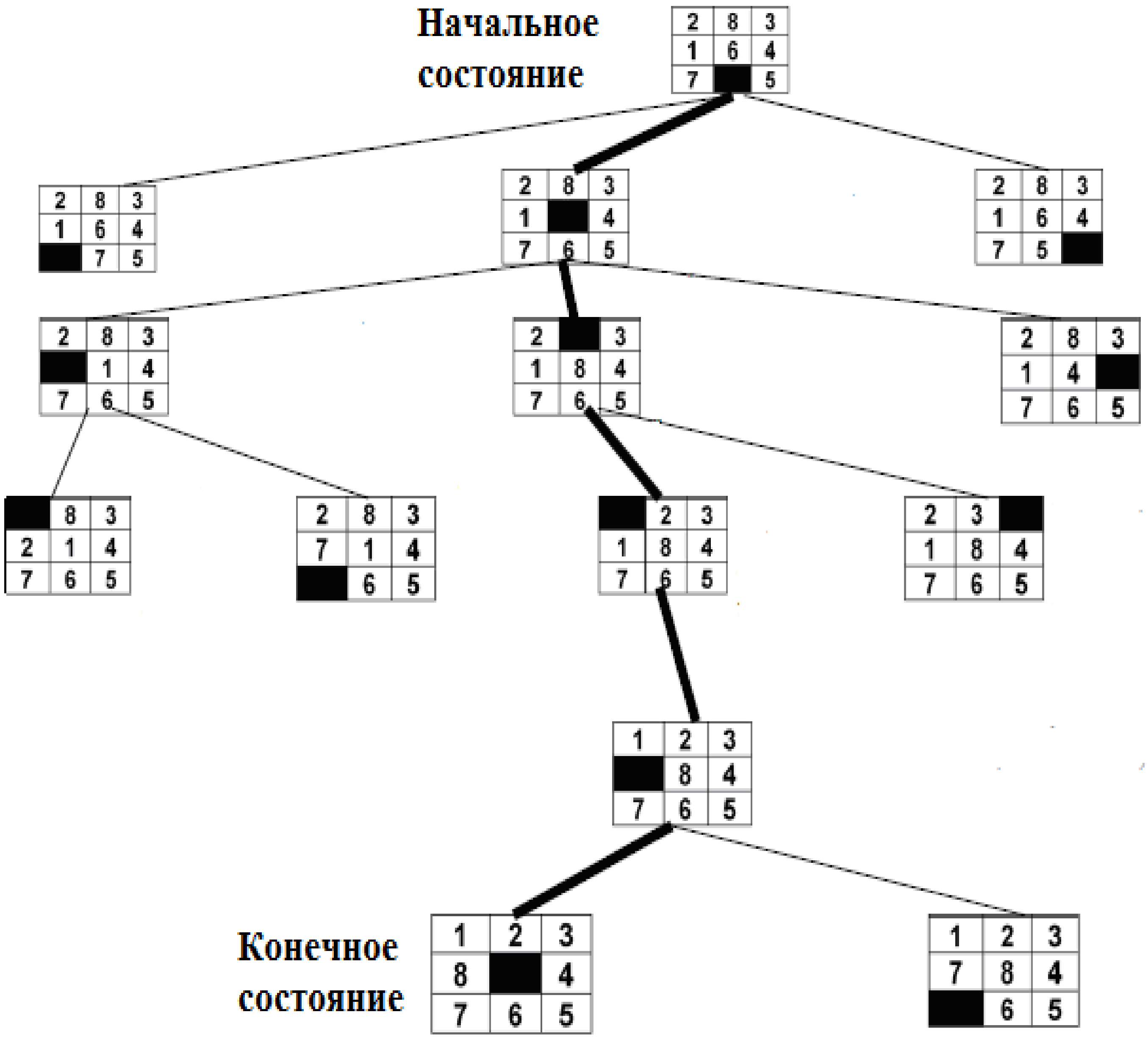
Процесс применения операторов к некоторой вершине с целью получения всех ее дочерних вершин называется ***раскрытием вершины*.**

Рассмотрим примеры формулировок (представлений) задач в пространстве состояний. Чтобы сформулировать задачу в такой форме требуется определить все элементы формулы (1)

Пример: задача перемещения дисков



Подграф поиска решения игры в восемь



В.Бондарев

Представление (описание) игры в восемь

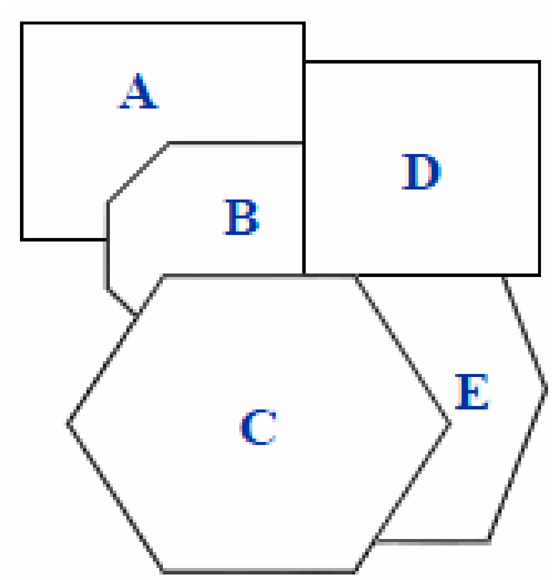
1. Состояние. Определяется местонахождением каждой из восьми фишек и пустой ячейки.
2. Начальное состояние. В качестве начального состояния может быть выбрано любое состояние.
3. Целевое состояние. Состояние с заданным упорядоченным расположением фишек.
4. Функция (оператор) преобразования состояний. Эта функция формирует допустимые состояния, которые являются результатом осуществления одного из четырех действий (теоретически возможных ходов: *влево, вправо, вверх, вниз*).
5. Целевая проверка. Сравнение текущего состояния и целевого состояния.

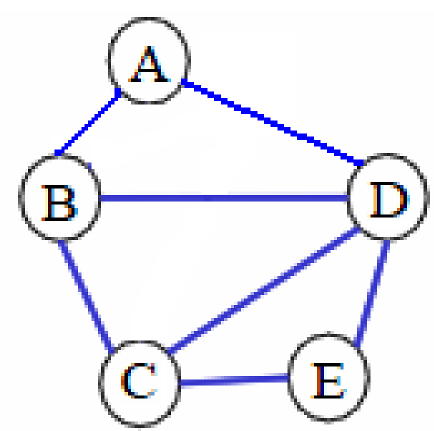
Задача о раскраске карты

Необходимо раскрасить плоскую карту, используя только 4 цвета. Соседние регионы на карте не должны раскрашиваться одним и тем же цветом.

Представление задачи

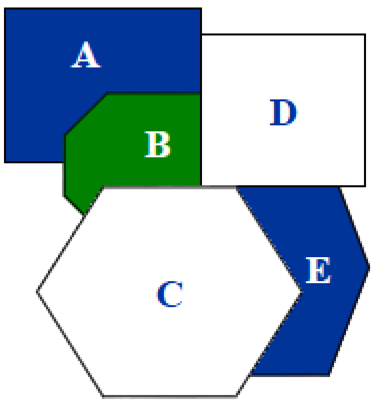
Представим карту в виде графа, узлы которого соответствуют регионам на карте, а ребра – выражают отношение соседства регионов.





Представление задачи о раскраске карты

Введем множество имен вершин **{v1,v2,…,vN}.**
Множество цветов (красок) **{c1,c2,c3,c4}.**
Состояние представим в виде **N** элементной записи, состоящей из цветов вершин. Вершина имеет цвет **x**, если ей не был назначен цвет. Пример состояния: **{c1,x,c1,c3,x,x,x…}**.



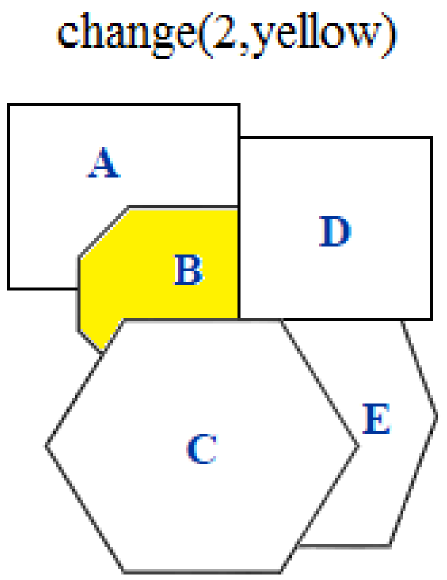
Состояние задачи для рисунка:

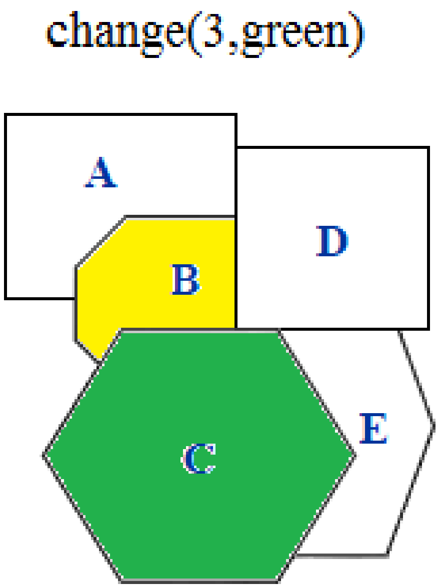
{blue,green,x,x,blue}

Начальное состояние: {x,x,…,x}.
Целевая проверка : все вершины имеют цвет и для каждой пары вершин **vi** и **vj**, которые являются смежными, состояние должно удовлетворять **color(i) ≠ color(j).**
Множество целевых состояний: множество состояний, удовлетворяющих указанным условиям

Представление задачи о раскраске карты

Функция преобразования состояний: change(i,c) – заменить цвет вершины (региона) **i** на **c**.

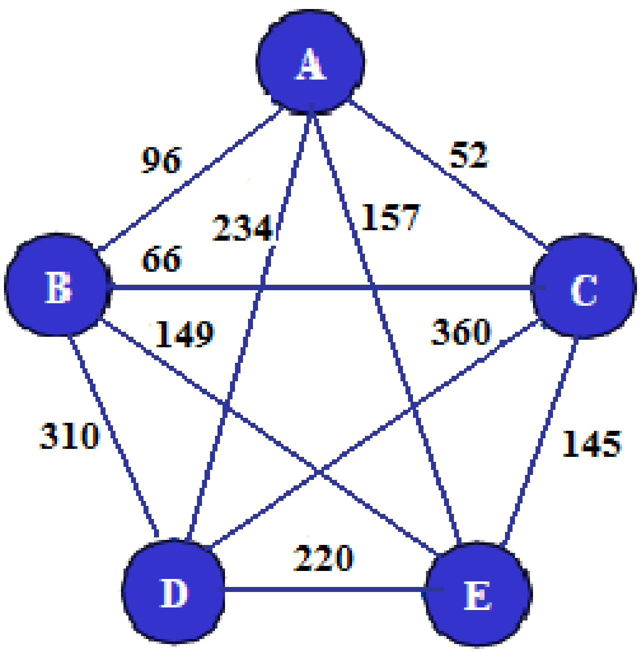




В.Бондарев

Представление задачи о коммивояжере

Коммивояжер должен посетить каждый из **N** заданных городов. Между каждой парой городов имеется путь. Нужно, отправляясь из стартового города, найти самый короткий путь, по которому коммивояжер по одному разу проходит через каждый из городов и затем возвращается в стартовый город.



Пусть **X** – мно-во из **N** городов, а **d(x,y)** – расстояние между городами **x** и **y**.

X={A,B,C,D,E}

Представление задачи о коммивояжере

S – мно-во состояний. **Состояние** представляется в виде пути Гамильтона (ни один город не повторяется) .

S={(x1,x2,…,xn) | n=1,…,N+1, xi ∈ X, xi ≠ xj, кроме i=1, j=N+1}

Функция преобразования состояний – это расширение пути Гамильтона:

Δ(x1,x2,…,xn) = {(x1,x2,…,xn,xn+1) | xn+1∈ X, xn+1 ≠ xi для всех 1 ≤ i ≤ n}

Множество целевых состояний включает все состояния (пути) длиной N+1. Среди этих путей надо найти кратчайший путь.

Представление задачи о каннибалах и миссионерах

Три миссионера и три каннибала находятся на левом берегу реки. Имеется лодка, вмещающая не более двух человек. Если на каком-то берегу каннибалов окажется больше, чем миссионеров, то они съедят миссионеров. Требуется найти безопасный план пересечения реки.

Состояние: **{M,K,B},** где **M** - кол-во миссионеров на левом берегу,**K**- количество каннибалов на левом берегу,**B**- местонахождения лодки **L** или **R**, **M>=K.**

Начальное состояние: **{3,3,L}.**

Целевое состояние: **{0,0,R}.**

Операторы: {M,K}, где **M**- кол-во миссионеров в лодке, **K** – кол-во каннибалов в лодке. Допустимые операторы: {1,0}; {2,0}; {1,1}; {0,1}; {0,2}.

Представление задачи о двух кувшинах

Дан кувшин с водой емкостью 5 литров и пустой кувшин емкостью 2 литра. Требуется наполнить второй кувшин 1 литром воды. Воду можно либо выливать, либо переливать из одного кувшина в другой.

Состояние: {X,Y}, где **X** – объем воды в первом кувшине; **Y**- объем воды во втором кувшине.

Начальное состояние: **{5,0}.**

Целевое состояние: **{*,1},** где * - означает любой объем.

Операторы:

{X,Y}→{0,Y} – вылить первый кувшин

{X,Y}→{X,0} – вылить второй кувшин

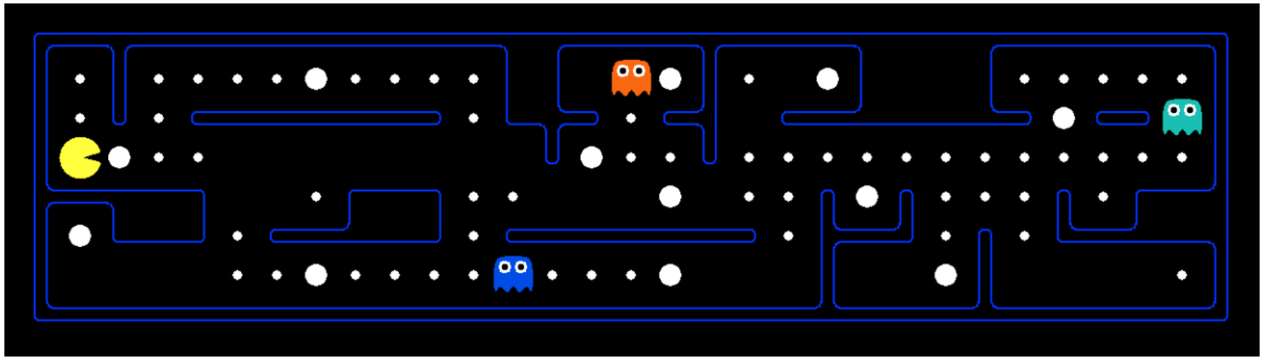
{X,2}→{X+2,0} – перелить 2-ой в 1-ый, X<=3.

{X,0}→{X-2,2} – перелить из 1-го во 2-ой, X>=2

{1,Y}→{0, Y+1} – перелить часть из 1-го во 2-ой, Y<2

Решение: {5,0} {3,2} {3,0} {1,2} {1,0} {0,1}

Представления игры Pacman



Рассмотрим 2 задачи поиска: нахождение пути в лабиринте и поедание всех точек. Ниже для этих задач перечислены: состояния, действия, функция-преемник и целевой тест

Нахождение пути:	Поедание всех точек (гранул):
- *состояние*: координаты (x, y) местоположения; - *действия*: Север, Юг, Восток, Запад; - *функция-преемник*: обновить только местоположение; - *проверка цели*: (x, y) = END?	- *состояние*: координаты (x, y) местоположения, логические точки (true/false); - *действия*: Север, Юг, Восток, Запад; - *функция-преемник*: обновить местоположение и логические значения точек; - *проверка цели*: все ли логические точки ложны?

Поиск решений в пространстве состояний

Методы поиска в пространстве состояний подразделяются на две группы:

- **неинформированный (слепой) поиск**:
  - поиск в ширину ;
  - поиск по критерию равной стоимости;
  - поиск в глубину (различные разновидности);
  - случайный поиск;
- **информированный (эвристический) поиск:**
  - локальные алгоритмы (hill-climbing);
  - глобальные алгоритмы поиска по первому наилучшему совпадению: жадный алгоритм; А-алгоритм.

В методах "слепого" поиска выполняется полный просмотр всего пространства состояний, что может приводить к проблеме ***комбинаторного взрыва.*** Для сокращения количества просматриваемых вариантов применяют методы информированного (направленного) поиска.

Критерии сравнения стратегий поиска

Полнота. Гарантируется ли обнаружение решения, если оно существует?

Оптимальность. Стратегию называют оптимальной, если она обеспечивает нахождение решения, которое не обязательно будет наилучшим, но известно, что оно принадлежит подмножеству решений, обладающих некоторыми заданными свойствами, согласно которым мы относим их к оптимальным.

Минимальность. Стратегию называют минимальной, если она гарантирует нахождение наилучшего решения, т.е. минимальность является более сильным случаем оптимальности.

Временная сложность. Время, необходимое для нахождения решения.

Пространственная сложность. Объем памяти, необходимый для решения задачи.