Решение задач и искусственный интеллект

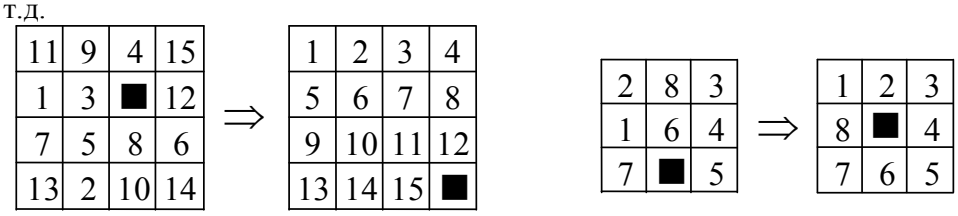
Двумя составными элементами процесса решения задач в теории искусственного интеллекта являются *представление* (формализация) задач и собственно решение – *поиск*. Мы рассмотрим два подхода к решению задач и, соответственно, два способа представления – подход с использованием пространства состояний и подход, основанный на редукции задач. Для обоих подходов описываются используемые алгоритмы поиска решения. Важной особенностью большинства этих алгоритмов является использование *эвристической* информации. *Эвристикой* обычно принято называть любое правило (стратегию, прием), существенно помогающее в решении некоторой задачи.

Представление задач в пространстве состояний

Основные понятия

Типичным представителем класса задач, для которых подходит представление в пространстве состояний, является головоломка, известная как игра в пятнадцать – см. рис. 1(а). В ней используется пятнадцать пронумерованных (от 1 до 15) подвижных фишек, расположенных в клетках квадрата 4×4. Одна клетка этого квадрата остается всегда пустой, так что одну из соседних с ней фишек можно передвинуть на место этой пустой клетки, изменив тем самым местоположение пустой клетки. Заметим, что более простым вариантом этой головоломки является квадрат 3×3 и восемь фишек на нем – пример соответствующей задачи показан на рис.1(б).

На рис.1(а) изображены две конфигурации фишек. В головоломке требуется преобразовать первую, или начальную, конфигурацию во вторую, или целевую конфигурацию. Решением этой задачи будет подходящая последовательность сдвигов фишек, например: передвинуть фишку 8 вверх, фишку 6 влево и т.д.



а) б)

Рис.1

Важной особенностью класса задач, к которому принадлежит рассмотренная головоломка, относится наличие в задаче точно определенной начальной ситуации и точно определенной цели. Имеется также некоторое множество операций, или ходов, переводящих одну конфигурацию в другую. Именно из таких ходов состоит искомое решение задачи, которое можно в принципе получить методом проб и ошибок. Действительно, отправляясь от начальной ситуации, можно построить конфигурации, возникающие в результате выполнения возможных в этой ситуации ходов, затем построить множество конфигураций, получающихся после применения следующего хода, и так далее – пока не будет достигнута целевая конфигурация.

Введем теперь основные понятия, используемые при формализации задачи в пространстве состояний. Центральным из них является понятие ***состояния***, характеризующего некоторый момент решения задачи. Например, для игры в пятнадцать (или в восемь) состояние – это просто некоторая конкретная конфигурация фишек.

Среди всех состояний задачи выделяются ***начальное состояние*** и ***целевое состояние***, в совокупности определяющие задачу, которую надо решить – примеры их приведены на рис.1.

Другим важным понятием является понятие ***оператора***, т.е. допустимого хода в задаче. Оператор преобразует одно состояние в другое, являясь по сути функцией, определенной на множестве состояний и принимающей значения из этого множества. Для игры в пятнадцать или в восемь удобнее выделить четыре оператора, соответствующие перемещениям пустой клетки (можно считать ее фишкой-«пустышкой») влево, вправо, вверх, вниз. В некоторых случаях оператор может оказаться неприменимым к какому-то состоянию: например, операторы сдвига вправо и вниз неприменимы, если пустая клетка расположена в правом нижнем углу. Значит, в общем случае оператор является частично определенной функцией отображения состояний.

В терминах состояний и операторов ***решение задачи*** есть определенная последовательность операторов, преобразующая начальное состояние в целевое. Решение задачи ищется в ***пространстве состояний*** – множестве всех состояний, достижимых из начального состояния при помощи заданных операторов. Например, в игре в пятнадцать или в восемь пространство состояний состоит из всех конфигураций фишек, которые могут быть образованы в результате возможных перемещений фишек.

Пространство состояний можно представить в виде направленного графа, вершины которого соответствуют состояниям, а дуги (ребра) – применяемым операторам. Указанные в виде стрелок направления соответствуют движению от вершины-аргумента применяемого оператора к результирующей вершине. Тогда решением задачи будет путь в этом графе, ведущий от начального состояния к целевому. На рис.2 показана часть пространства состояний для игры в пятнадцать. Каждая вершина соответствует некоторой конфигурации фишек. Все дуги между вершинами являются двунаправленными, поскольку в этой головоломке для любого оператора есть обратный ему (точнее, множество операторов состоит из двух пар взаимно-обратных операторов: влево-вправо, вверх-вниз).

Пространства состояний могут быть большими и даже бесконечными, но в любом случае предполагается счетность множества состояний.

Таким образом, в подходе к решению задачи с использованием пространства состояний задача рассматривается как тройка $\mathbf{(S_I , O , S_G)}$, где

$\mathbf{S_I}$ – начальное состояние;

O – конечное множество операторов, действующих на не более чем счетном множестве состояний;

$\mathbf{S_G}$ – целевое состояние.

Дальнейшая формализация решения задачи с использованием пространства состояний предполагает выбор некоторой конкретной *формы* описания состояний задачи. Для этого могут применяться любые подходящие структуры – строки, массивы, списки, деревья и т.п. Например, для игры в пятнадцать или восемь наиболее естественной формой описания состояния будет список положений фишек или же двумерный массив. Заметим, что от выбора формы описания состояния зависит в общем случае сложность задания операторов задачи, которые должны быть также определены при формализации задачи в пространстве состояний.

Если для игры в пятнадцать средством формализации выступает язык программирования Лисп или Паскаль, то операторы задачи могут быть описаны в виде четырех соответствующих функций языка. При использовании же продукционного языка, эти операторы задаются в виде правил продукций вида: «исходное состояние → результирующее состояние».

В рассмотренных выше примерах (игры в пятнадцать и восемь) искомое целевое состояние задавалось явно, т.е. известно было местоположение каждой фишки в целевой конфигурации. В более сложных случаях игры может быть несколько целевых состояний, либо же целевое состояние может быть определено неявно, т.е. охарактеризовано некоторым свойством, например, как состояние, в котором сумма номеров фишек в верхнем ряду не превосходит 10. В подобных случаях свойство, которому должно удовлетворять целевое состояние, должно быть описано исчерпывающим образом, к примеру, путем задания булевской функции, реализующей проверку нужного свойства состояния задачи.

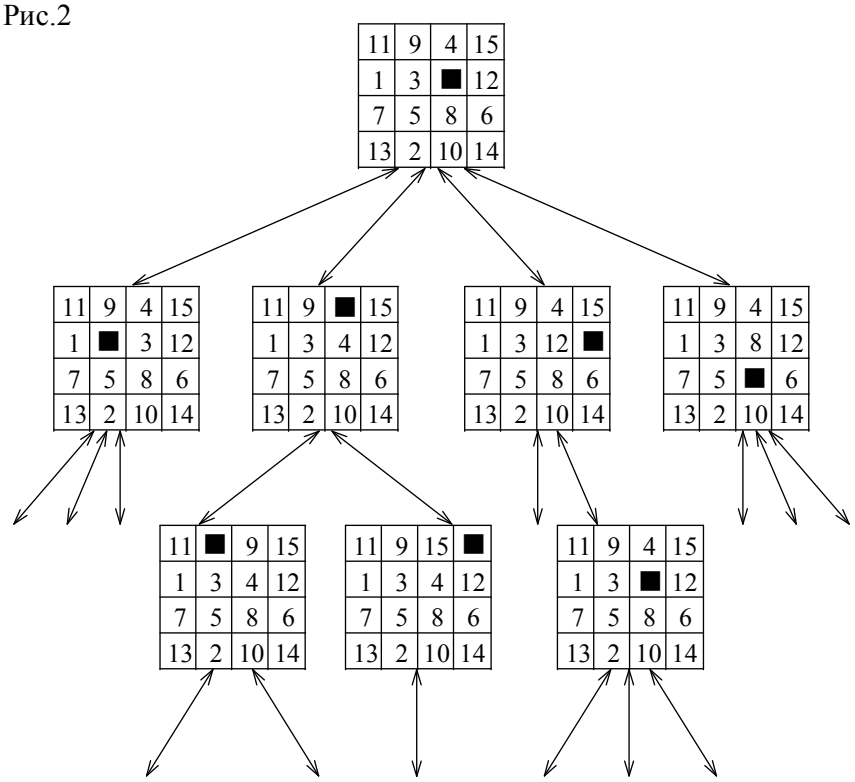
Итак, для представления задачи в пространстве состояний необходимо определить следующее:

- форму описания состояний задачи и описание начального состояния;
- множество операторов и их воздействий на описания состояний;
- множество целевых состояний или же описание их свойств.

Перечисленные составляющие задают неявно граф-пространство состояний, в котором требуется провести поиск решения задачи. Заметим попутно, что в отличие от такого неявного способа задания графа, при явном способе задания все вершины и дуги графа должны быть перечислены, например, с помощью таблиц.

Решение задачи в пространстве состояний подразумевает просмотр неявно заданного графа, для чего необходимо преобразование в явную форму достаточно большой его части, включающей искомую целевую вершину. Действительно, просмотр осуществляется как последовательный ***поиск***, или ***перебор*** вершин, в ***пространстве состояний***. В исходной точке процесса к начальному состоянию применяется тот или иной оператор и строится новая вершина-состояние, а также связывающие ее с корневой вершиной дуги. На каждом последующем шаге поиска к одной из уже полученных вершин-состояний применяется допустимый оператор и строится еще одна вершина графа и связывающие дуги. Этот процесс поиска продолжается до тех пор, пока не будет построена вершина, соответствующая целевому состоянию.

Рис.2

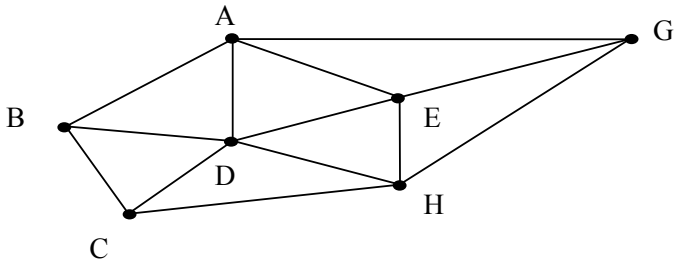


Примеры пространств состояний

Разберем два характерных примера представления в пространстве состояний, показывающих, что такое представление возможно для различных типов задач. Подчеркнем заранее, что предлагаемые ниже представления, хотя и являются достаточно естественными, не являются единственно допустимыми в этих задачах, возможны и другие варианты.

Вообще, от выбора представления, т.е. рассмотренных выше составляющих, зависит размер пространства состояний, а значит, и эффективность поиска в нем. Очевидно, желательны представления с малыми пространствами состояний, но нахождение сужающих пространство поиска удачных представлений требует обычно некоторого дополнительного анализа решаемой задачи.

Рассмотрим формализацию в пространстве состояний известной *задачи о коммивояжере* (представляющей классическую переборную проблему). Коммивояжер, располагая картой дорог, соединяющей 7 городов (рис. 3), должен построить свой маршрут так, чтобы выехав из города A, посетить каждый из других шести городов B, C, D, E, H, G в точности по одному разу и затем вернуться в исходный город. В другом, более сложном варианте задачи требуется также, чтобы маршрут имел минимальную протяженность.



Р и с . 3

Состояние решаемой задачи можно задать как список городов, которые уже проехал коммивояжер к текущему моменту. Тогда возможным состояниям соответствуют списки из элементов A, B, C, D, E, H, G без повторений, исключение составляет только город A, он может встретиться в списке дважды – в начале списка и его конце. Пример списка-состояния – (A B C H). Начальное же состояние определяется как список (A), а целевое – как любой допустимый список, начинающийся и кончающийся элементом A. Для определенных таким образом состояний задачи операторы задачи могут соответствовать перемещениям между городами – получаем таким образом 13 операторов.

Рассмотрим теперь широко известную *задачу об обезьяне и банане*, простейшую формулировку которой мы и рассмотрим. В комнате находятся обезьяна, ящик и связка бананов, которая подвешена к потолку настолько высоко, что обезьяна может до нее дотянуться, только встав на ящик. Нужно найти последовательность действий, которая позволит обезьяне достать бананы. Предполагается, что обезьяна может ходить по комнате, двигать по полу ящик, взбираться на него и хватать бананы.

Ясно, что описание состояния этой задачи должно включать следующие сведения: местоположение обезьяны в комнате – в горизонтальной плоскости пола и по вертикали (т.е. на полу она или на ящике), местоположение ящика на полу и наличие у обезьяны бананов. Все это можно представить в виде четырехэлементного списка *(ПолОб, ВертОб, ПолЯщ, Цель),* где

ПолОб – положение обезьяны на полу (это может быть двухэлементный вектор координат);

ПолЯщ – положение обезьяны и ящика на полу;

ВертОб – это константа П или Я в зависимости от того, где находится обезьяна, на полу или на ящике;

Цель – это константа 0 или 1 в зависимости от того, достала ли обезьяна бананы или нет.

Зафиксируем также как константы три следующие точки в плоскости пола:

$T_{О}$ – точка первоначального местоположения обезьяны;

$T_{Я}$ – точка первоначального расположения ящика;

$T_{Б}$ – точка пола, расположенная непосредственно под связкой бананов.

Тогда начальное состояние задачи описывается списком ($T_{О}$, П, $T_{Я}$, 0), а целевое состояние задается как любой список, последний элемент которого – 1.

Естественно определить операторы в этой задаче в соответствии с возможными действиями обезьяны:

1) *Перейти (W)* – переход обезьяны к точке W горизонтальной плоскости пола;
2) *Передвинуть (V)* – передвижение обезьяной ящика в точку V пола;
3) *Взобраться* – обезьяна взбирается на ящик;
4) *Схватить* – обезьяна хватает связку бананов.

Условия применимости и действие этих операторов легко определить в виде правил продукций вида: *аргумент оператора* $\rightarrow$ *результат оператора,*

причем

X, Y, Z, W, V обозначают переменные:

1. *Перейти (W)* : (X, П, Y, Z) $\rightarrow$ (W, П, Y, Z)
2. *Передвинуть (V)* : (X, П, X, Z) $\rightarrow$ (V, П, V, Z)
3. *Взобраться* : (X, П, X, Z) $\rightarrow$ (X, Я, X, Z)
4. *Схватить:* ($T_{Б}$, Я,$T_{Б}$, 0) $\rightarrow$ ($T_{Б}$, Я, $T_{Б}$, 1)

Будем считать, что для решения задачи значимы лишь вышеупомянутые точки пола $T_{О}$, $T_{Я}$, $T_{Б}$, тогда получим пространство состояний задачи, изображенное на рис. 4. Это пространство содержит только 13 состояний, дуги графа-пространства промаркированы порядковым номером применяемого оператора. Пространство содержит четыре цикла хождения обезьяны между тремя значимыми точками (с ящиком или без него). В пространстве есть также две тупиковые ветви – когда обезьяна залезает на ящик, но не под связкой бананов. Жирными дугами (стрелками) показан решающий путь, состоящий из четырех операторов: ***Перейти ($T_{Я}$); Передвинуть($T_{Б}$); Взобраться; Схватить***.

Рис. 4