Сведение задач к подзадачам

Броварь Ирина

ВМК МГУ 2011

GPS

GPS (General Problem Solver) — известная программа, предложенная Алэном Ньюэллом, Клиффом Шоу и Гербертом Саймоном в конце 50-х гг. (одна из самых первых программ искусственного интеллекта) и способная решать однотипным способом такие непохожие задачи, как подсчет интеграла, логические головоломки, доказательство теорем методом исчисления предикатов, грамматический анализ фразы.

GPS

GPS (General Problem Solver) — известная программа, предложенная Алэном Ньюэллом, Клиффом Шоу и Гербертом Саймоном в конце 50-х гг. (одна из самых первых программ искусственного интеллекта) и способная решать однотипным способом такие непохожие задачи, как подсчет интеграла, логические головоломки, доказательство теорем методом исчисления предикатов, грамматический анализ фразы.

Входные данные

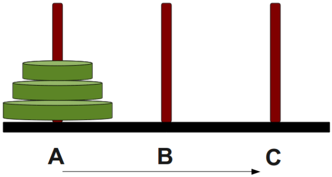
- Исходный объект
- Конечный объект
- Множество операторов

Выходные данные

Последовательность операторов, приводящая исходный объект к конечному.

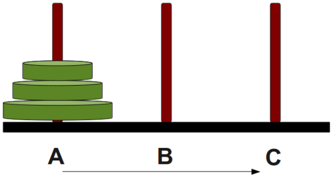
Ханойские башни — 1

Исходный объект:

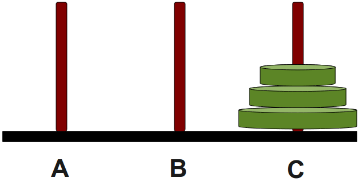


Ханойские башни — 1

Исходный объект:

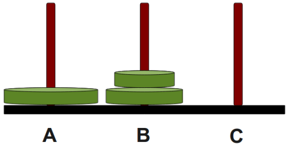


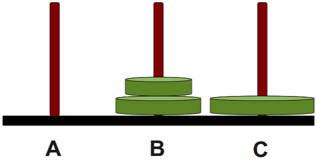
Конечный объект:



Ханойские башни — 2

Обязательный шаг:





Описание задач

Любая задача поиска в пространстве состояний может быть представлена в виде:

- Множество S начальных состояний
- Множество F операторов, отображающих описания состояний в описания состояний
- Множество G целевых состояний

Тройка (S, F, G) описывает задачу.

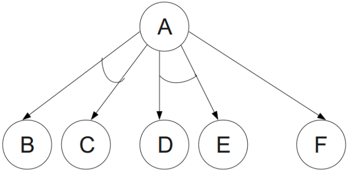
Операторы сведения задач к подзадачам

Описание задачи $\Rightarrow$ описания подзадач (дочерние задачи).
Подзадачи могут быть описаны тройками (S', F', G') из подмножеств исходных множеств.
Это преобразование таково, что решение всех дочерних задач обеспечивает решение исходной задачи.
Применение каждого такого оператора порождает альтернативные множества подзадач.

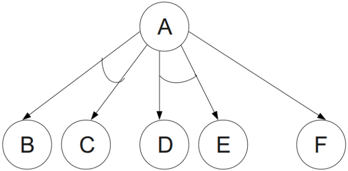
Цель:

получение элементарных задач (задачи, решения которых очевидны)

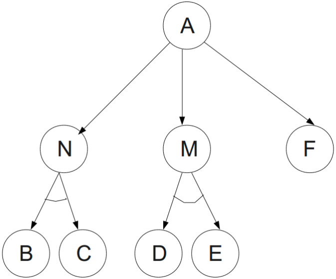
И/ИЛИ графы



И/ИЛИ графы



N, M, F — альтернативные подзадачи



Поиск на И/ИЛИ графах — 1

Разрешимые вершины

- Заключительные вершины (соответствующие элементарным задачам) разрешимы
- Вершина, не являющаяся заключительной и имеющая дочерние вершины типа "ИЛИ разрешима $\Leftrightarrow$ разрешима по крайней мере одна из ее дочерних вершин
- Вершина, не являющаяся заключительной и имеющая дочерние вершины типа "И разрешима $\Leftrightarrow$ разрешимы все ее дочерние вершины

Поиск на И/ИЛИ графах — 2

Неразрешимые вершины

- Вершины, не являющиеся заключительными и не имеющие дочерних вершин, неразрешимы
- Вершина, не являющаяся заключительной и имеющая дочерние вершины типа "ИЛИ неразрешима $\Leftrightarrow$ неразрешимы все ее дочерние вершины
- Вершина, не являющаяся заключительной и имеющая дочерние вершины типа "И неразрешима $\Leftrightarrow$ неразрешима по крайней мере одна из ее дочерних вершин

Решающий граф — подграф из разрешимых вершин, показывающий, что начальная вершина разрешима.

Поиск на И/ИЛИ графах — 3

Общий алгоритм

1. Заходим в начальную вершину
2. Строим множество дочерних вершин
3. Для каждой дочерней вершины запускаем алгоритм
4. В соответствии с разрешимостью/неразрешимостью дочерних вершин, устанавливаем разрешимость/неразрешимость вершины.

Алгоритмы обхода дерева

- Поиск в ширину
- Поиск в глубину

Ключевые операторы

Как свести задачу к более простым задачам?

Если известны основные промежуточные состояния $g_1, g_2, ..., g_N$ необходимые в любом решении, то можно свести начальную задачу (S, F, G) к множеству подзадач $(S, F, g_1), (g_1, F, g_2), ..., (g_N, F, G)$ которые можно решать в любом порядке.

Ключевые операторы — операторы, находящиеся в решающей цепочке операторов для каждого из решений.

$$(S, F, G) \Rightarrow (f)(S, F, G_f) \cup (g, F, f(g)) \cup (f(g), F, G)$$

Различия

Различия для (S, F, G) — частичный список причин, по которым элементы множества S не удовлетворяют тем условиям, которым должны удовлетворять элементы множества G.

Различие $\Rightarrow$ множество операторов, устраняющих это различие Такие операторы могут быть ключевыми с большой вероятностью.

Обезьяна и бананы — 1 (Описание задачи)

Пространство состояний

(w, x, y, z)

w — координаты обезьяны в горизонтальной плоскости

x — 1, если обезьяна находится на ящике или 0, если нет

y — координаты ящика в горизонтальной плоскости

z — 1, если обезьяна достала бананы или 0, если нет

Обезьяна и бананы — 2 (Описание задачи)

Начальное состояние

$(a, 0, b, 0)$

Конечное состояние

$(c, 1, c, 1)$, где c — точка, в которой висят бананы

Операторы

- $f_1(w, 0, y, z) \Rightarrow (u, 0, y, z)$ подойти
- $f_2(w, 0, w, z) \Rightarrow (v, 0, v, z)$ передвинуть
- $f_3(w, 0, w, z) \Rightarrow (w, 1, w, z)$ взобраться
- $f_4(c, 1, c, 0) \Rightarrow (c, 1, c, 1)$ схватить

Обезьяна и бананы — 3

Исходная задача — $(a, 0, b, 0) \Rightarrow (w, x, y, 1)$
Различие — последнее число не равно 1
Ключевой оператор — f_4
Условия применения — $(c, 1, c, 0)$

Обезьяна и бананы — 3

Исходная задача — $(a, 0, b, 0) \Rightarrow (w, x, y, 1)$
Различие — последнее число не равно 1
Ключевой оператор — f_4
Условия применения — $(c, 1, c, 0)$

Подзадача — $(a, 0, b, 0) \Rightarrow (c, 1, c, 0)$
Различия —

- Ящик не находится в точке c
- Обезьяна не находится в точке c
- Обезьяна не на ящике

Ключевые операторы — $f_2(c)$, $f_1(c)$, f_3

Обезьяна и бананы — 3

Первый вариант — f_2 (передвинуть (c))

Условия применения — $(w, 0, w, z)$
Подзадача — $(a, 0, b, 0) \Rightarrow (b, 0, b, 0) \Rightarrow (c, 0, c, 0) \Rightarrow (c, 1, c, 0)$
Различие — Обезьяны нет в точке b
Ключевые операторы — $f_1(b)$

Последовательность операторов, решающих исходную задачу:
подойти (b), передвинуть (c), взобраться, схватить)

Символьное интегрирование

Исходный объект

Неопределенный интеграл

Конечный объект

Формула посчитанного интеграла без учета константы

Множество операторов

- Таблица простых интегралов
- Правила интегрирования
- Алгебраические подстановки
- Тригонометрические подстановки
- Деление числителя на знаменатель
- Дополнение до полного квадрата

Источники

- Н. Нильсон, «Искусственный интеллект»
- Ж.-Л Лорьер, «Системы искусственного интеллекта»