

Различия и ключевые операторы

В заключение целесообразно обсудить ряд общих идей, на которых может быть основана редукция задач. К их числу прежде всего относится идея выделения так называемых **ключевых операторов** и **различий** состояний. Будем далее предполагать, что задачи и подзадачи описываются в пространстве состояний как тройки вида (S_I, O, S_G) , где S_I и S_G – соответственно начальное и целевое состояния, или же как двойки (S_I, S_G) , если предполагать неизменным при поиске множество O операторов преобразования состояний.

Часто при поиске в пространстве состояний нетрудно обнаружить один оператор, который обязательно должен входить в решение задачи (по сути, применение этого оператора есть необходимый шаг решения этой задачи). Такой оператор и называется ключевым оператором в пространстве состояний. К примеру, в задаче о пирамидке ключевым оператором был оператор переноса на нужный колышек самого большого диска 3.

Ключевой оператор может быть использован для следующего способа сведения исходной задачи к подзадачам. Пусть

Op – найденный в пространстве состояний задачи ключевой оператор ($Op \in O$);

S_{Op} – множество состояний, к которым применим ключевой оператор Op ;

$Op(s)$ – состояние, полученное в результате применения ключевого оператора Op к состоянию s ($s \in S_{Op}$).

Тогда исходную задачу можно свести к трем подзадачам:

- первая задача (S_I, S_{Op}) состоит в поиске пути от начального состояния исходной задачи к одному из состояний, к которому применим ключевой оператор Op ;
- вторая задача является элементарной, она заключается в применении этого ключевого оператора;
- третья задача $(Op(s), S_G)$ состоит в поиске пути от состояния, полученного применением ключевого оператора, к целевому состоянию исходной задачи.

Порядок решения перечисленных задач существен (третья задача не может быть решена раньше первой и второй, также как вторая – раньше первой). Для решения первой и третьей задач может быть применен опять метод редукции с помощью ключевого оператора, или же их решение может быть найдено непосредственным поиском в пространстве состояний.

Для большинства задач не удается всегда однозначно выделить ключевой оператор, гораздо чаще удается найти множество *операторов-кандидатов* в ключевые (т.е. операторов, с большой вероятностью могущих стать ключевыми). Таким образом, в общем случае необходим процесс перебора операторов-кандидатов, каждый из которых образует свое множество результирующих задач (этот перебор и означает поиск на И/ИЛИ-графе задачи).

Самый важный вопрос при таком способе редукции задач состоит в том, как найти кандидаты в ключевые операторы. Один из способов, предложенных и опробованных впервые в одной из наиболее известных систем искусственного интеллекта “General Problem Solver” - GPS (А.Ньюэлл, Г.Саймон, Дж.Шоу – 1957 г.), заключается в выявлении *различий* для начального и целевого состояний задачи.

Различие легче всего формализовать как несоответствие различных элементов описаний начального и целевого состояний.

К примеру, в задаче об обезьяне и банане различием можно считать неравенство соответствующих элементов списков, описывающих два состояния. Тогда при сравнении состояний $(T_0, П, T_я, 0)$ и $(T_я, П, T_0, 1)$ выявляются три различия – соответственно в первых, третьих и четвертых элементах списков.

С каждым различием в системе GPS был связан один или несколько операторов, призванных устранять или уменьшать это различие. Эти операторы и являлись по сути кандидатами в ключевые. На каждом этапе работы система определяла различия между текущим состоянием (объектом) задачи и целевым состоянием (объектом), а затем выбирала и пыталась применить оператор для уменьшения найденного различия. В общем случае операторы включали в себя *предусловия* (условия применимости), выполнение которых было необходимо для их применения, в этом случае GPS сводила исходную задачу к задаче достижения нужного условия.

В задаче об обезьяне и банане естественно связать различия и операторы-кандидаты в ключевые следующим образом:

- Различие в первом элементе списка-описания состояния (положение обезьяны в плоскости пола) – операторы *Перейти* и *Передвинуть*.
- Различие во втором элементе (положение обезьяны по вертикали) – оператор *Взобраться*.
- Различие в третьем элементе (положение ящика) – оператор *Передвинуть*.
- Различие в четвертом элементе (содержимое руки обезьяны) – оператор *Схватить*.

Ясно, что для реализации рассмотренных идей в виде алгоритма или программы должна быть, во-первых, специальная процедура сравнения описаний состояний и вычисления различий. Во-вторых, необходима процедура, связывающая ключевые операторы с возможными различиями (в GPS использовалась таблица связей различие-оператор). Последняя процедура должна также устанавливать порядок устранения различий в случаях, когда выявлено несколько различий и возможно применение нескольких ключевых операторов. Различия должны быть упорядочены по степени их существенности, значимости для конечного решения исходной задачи. Система GPS начинала с попытки обработки более серьезных и трудно устранимых различий, переходя затем к более легким.

В задаче об обезьяне и банане приоритет (существенность) различий можно установить, например, следующим образом: различие в четвертом, затем во втором, затем в третьем и первом элементе описания состояния задачи. Такой же приоритет может быть и у операторов, уменьшающих эти различия.

Важно, что воплощаемая в указанных процедурах информация является специфической, зависящей от конкретной задачи, т.е. эвристической проблемно-ориентированной информацией. Одной из слабостей применяемого в системе GPS подхода было то, что процедуры определения различий и уменьшающих их операторов должны были быть отдельно реализованы для каждой конкретной задачи (или для очень узкой предметной области, включающей несколько видов задач), в противном случае снижалась эффективность решения задач.

Подчеркнем, что основной механизм системы GPS не был проблемно-ориентированным: он представлял собой реализацию универсального эвристического метода решения задач, часто применяемого человеком, и известного как *анализ целей и средств (means-ends analysis)*. Ключевая идея этой эвристики такова:

- поиск различий между тем, что дано в поставленной задаче, и тем, что надо получить;
- последовательное устранение найденных различий с помощью подходящих средств-операций.

Работая в соответствии с этой эвристикой, GPS применяла несколько схем редукции задач (*Методов*), и на основе выявления различий между объектами задачи и применения уменьшающих эти различия операторов рекурсивно формировала систему (дерево) задач-целей (подзадач).

Краткое описание схемы работы системы GPS

Проблемная среда в системе GPS описывается с использованием таких понятий/терминов:

Объекты (элементы проблемной среды)

Различия (между Объектами)

Операторы (способы преобразования Объектов)

Цель

При планировании решения задачи в системе GPS используются:

Три основных *Метода* (не зависящих от конкретной предметной области):

1. **Преобразовать** один *Объект* в другой: $A \rightarrow B$

а) Сравнить A с B , найти $D = (A - B)$

если $D = 0$, то **FIN (успех)**

б) **ПОДЦЕЛЬ: Уменьшить Различие D**

если это не удастся, то **FIN (неудача)**, иначе: найдется A' (нет *Различия D* с B)

в) **ПОДЦЕЛЬ: Преобразовать $A' \rightarrow B$**

если это не удастся, то **FIN (неудача)**, иначе **FIN (успех)**

2. **Уменьшить Различие** между двумя *Объектами*: $D = (A - B)$

а) Найти оператор Q , подходящий для уменьшения *Различия D*

если это не удастся, то **FIN (неудача)**

б) Предварительная проверка применимости *Оператора*

если *Оператор* эту проверку не прошел, то **FIN (неудача)**

в) **ПОДЦЕЛЬ: Применить Оператор Q (A), результат A' , FIN (успех)**

3. **Применить Оператор** к *Объекту*: $Q(A)$

а) Найти $D = (\text{prec}(Q) - A)$

если $D = 0$, то результат A'' , **FIN (успех)**

б) **ПОДЦЕЛЬ: Уменьшить Различие D**

если это не удастся, то **FIN (неудача)**, иначе: найдется A' (нет *Различия D* с $\text{prec}(Q)$)

в) **ПОДЦЕЛЬ: Применить Оператор Q (A'), результат A'' ,**

если это не удастся, то **FIN (неудача)**, иначе **FIN (успех)**

Для использования системы GPS в конкретной предметной области необходимо описать Объекты этой ПО и множество учитываемых Различий между Операторами, а также установить связи между Различиями и Операторами (для каждого Оператора указать, какие Различия он может устранить). Эти связи можно задать с помощью таблицы (демонстрационный пример):

Операторы \ Различия	Op ₁	Op ₂	Op ₃	Op ₄	...	Op _N
D ₁	+					
D ₂	+		+			
D ₃		+				
D ₄		+				
...						
D _M			+			+

Символ «+» в позиции $\langle i, j \rangle$, что Op_j может устранить различие D_i.

Предметная область (в которой реально проводились эксперименты с системой GPS) – «Преобразование логических выражений»:

Примеры *Различий*:

D₁ – в формуле не хватает символов (добавить символ)

D₂ – в формуле есть лишние символы (вычеркнуть символ)

D₃ – в формуле другая связка (изменить связку)

D₄ – в формуле другой знак выражения (изменить знак)

Примеры *Операторов*:

Op₁ : $A \& A \rightarrow A$; $A \vee A \leftrightarrow A$

Op₂ : $A \vee B \leftrightarrow (\sim A \& \sim B)$

Op₃ : $A \& B \rightarrow A$; $A \& B \leftrightarrow B$ [относится только к основному выражению]

Правила предварительной проверки применимости оператора:

- Одинакова ли главная связка?
- Не слишком ли «велик» Оператор? $\{ (A \vee B) \& (A \vee C) \rightarrow A \vee (B \& C) \}$
- Не слишком ли «прост»/неспецифичен Оператор? $\{ A \rightarrow A \& A \}$
- Удовлетворены ли побочные условия? $\{ \text{[относится только к основному выражению]} \}$

Интеллектуальные роботы

Интеллектуальный робот (его следует отличать от также иногда называемых роботами технических систем, типа систем точечной сварки и т.п.) – программно-аппаратный комплекс, оснащенный **акцепторами** (датчиками о состоянии проблемной среды) и **эффекторами** (средствами воздействия на эту среду, в частности, средствами передвижения), в состав которого входит система ИИ, способная к планированию действий робота в среде.

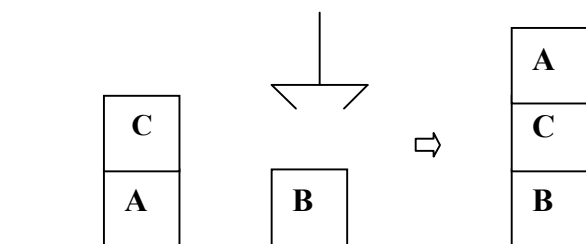
В наши дни ведутся работы по «интеллектуализации» технических/промышленных роботов.

Часто требуется возможность автономного функционирования робота в проблемной среде (например, в среде агрессивной, в которой человек находиться не может).

Иногда предполагается возможность передачи роботу (человеком-оператором) управляющих команд.

В свое время (70-е гг. XX века) задача создания интеллектуальных роботов рассматривалась как универсальная задача-рамка для исследований в области ИИ. Действительно, помимо таких проблем как представление знаний, планирование решения при создании роботов приходится ставить и решать задачи обработки изображений, управления эффекторами, не возникающие в случае более традиционных систем ИИ (решателей интеллектуальных задач).

Пример: ПЛАНИРОВАНИЕ ДЕЙСТВИЙ РОБОТА В «МИРЕ КУБИКОВ»



Начальное состояние:

CL (C), CL (B), ON (C, A), ONT (A), ONT (B),
HEMP, HOLD (x)

Целевое состояние:

ON (C, B) & ON (A, C) [& ONT (B)]

Сложность достижения «целей»:

HEMP	1
CL (x)	2
ONT (x)	2
HOLD (x)	2
ON (x, y)	3

Операторы:

Pickup (x)

prec: ONT (x), CL (x), HEMP
 res-: ONT (x), CL (x), HEMP
 res+: HOLD (x)

Putdown (x)

prec: HOLD (x)
 res-: HOLD (x)
 res+: ONT (x), CL (x), HEMP

Stack (x, y)

prec: HOLD (x), CL (y)
 res-: HOLD (x), CL (y)
 res+: ON (x, y), CL (x), HEMP

Unstack (x, y)

prec: ON (x, y), CL (x), HEMP
 res-: ON (x, y), CL (x), HEMP
 res+: HOLD (x), CL (y)

Планирование способов достижения целевого состояния: ON (C, B) & ON (A, C)

Иерархическое планирование:

1. Планирование на высшем (3) уровне сложности:

{ Stack (C, B), Stack (A, C) } – предусловия пока не рассматриваем

2. Учет предусловий (планирование на других уровнях сложности):

HOLD (C), CL (A) – res+ оператора Unstack (C, A), CL (B) – выполняется [начальное состояние]

Stack (C, B)

HOLD (A) – res+ оператора Pickup (A), CL (C) – выполняется [res+ оператора Stack (C, B)]

Stack (A, C)

3. Окончательный план: { Unstack (C, A), Stack (C, B), Pickup (A), Stack (A, C) }

ИНТЕЛЛЕКТУАЛЬНЫЕ РОБОТЫ и ПРОМЫШЛЕННЫЕ РОБОТЫ

История. «Телетанки» – 30-е гг. XX века (предложил маршал Тухачевский);

Германия (во второй мировой войне) – около 7000 шт. Управление по радио. Проблемы.

Чернобыльская авария. Высокий уровень радиации выдерживали только отечественные роботы.

Наше время. Луноход (часто можно управлять с Земли), Марсоход (более актуально).

Подводные роботы (Балтийское море: разминирование, поиск химического оружия).

Война в Ираке (воздушная разведка – несколько сотен беспилотных ЛА).

Основные военные приложения: разминирование, разведка, санитарная служба (поиск раненых, экстренная медицинская помощь, эвакуация с поля боя).

Прогноз. Через 20 лет роботы – основной вид вооружений.

В последние годы понятие *интеллектуальный робот* в какой-то степени вытесняется близким понятием *интеллектуальный агент*.

Интеллектуальные Агенты

Термин *Интеллектуальный Агент*, ставший популярным около 10 лет назад, трактуется различными исследователями по-разному.

Обычно предполагается, что *Интеллектуальный Агент*:

- действует автономно или совместно с другими компьютерными/интеллектуальными системами;
- выполняет шаблонные предписанные действия и/или действия, требующие активности и учета состояния окружающей среды;
- в той или иной степени способен к обучению, корпоративным действиям; мобилен.

В работах, посвященных *Интеллектуальным Агентам*, указываются, в частности, следующие сферы их применения (и соответствующие технологии):

агенты, поддерживающие интеллектуальный пользовательский интерфейс;
 мульти-агентные системы (и технологии «распределенного искусственного интеллекта»);
 мобильные агенты (в том числе реализованные программно), которые могут общаться между собой и перемещаться в своем специфическом окружении (в частности, в компьютерных сетях).

Рассуждения в условиях неопределенности

«Любая традиционная логика обычно предполагает использование точных символов. Поэтому она применима не к земной жизни, а лишь к воображаемому небесному существованию» Б. Рассел.

Традиционная логика: из корректных предпосылок с помощью обоснованных правил вывода → новые гарантированно корректные заключения.

С помощью правил вывода количество известной информации монотонно увеличивается.

Реально: неопределенность, мера/степень доверия, мера доверия и выводы меняются в процессе работы.

Немонотонные рассуждения:

Заключения/выводы могут пересматриваться.

Допускается признание истинности некоторого факта на основе «здравого смысла» (как устроен мир обычно).

Допускается использование неточных и противоречивых данных.

Подходы:

1.Расширение логических формализмов.

Логическая абдукция.

Абдуктивный вывод: $P \rightarrow Q$, если имеет место Q , то можно сделать вывод, что имеет место P ($Q \Rightarrow P$).

Пример:

если двигатель не вращается и фары не горят (Q), то проблема в аккумуляторе или проводке (P)
 [это неверно; возможно (но маловероятно), что поврежден стартер и перегорели все электролампы]

$\langle P \rightarrow Q, Q \Rightarrow P \rangle$

Обратное верно:

если проблема в аккумуляторе или проводке (P), то двигатель не вращается и фары не горят (Q)

$\langle P \rightarrow Q, P \Rightarrow Q \rangle$

2.Неточный вывод на основе фактора уверенности / нечеткие рассуждения / теория Демпстера-Шефера

3.Стохастические подходы (основанные на теореме Байеса).

Расчет вероятности верности гипотезы в условиях неполной и/или непредставительной выборки (на основе наблюдений известна лишь часть информации о событиях).