Введение

В бурно развивающейся науке «искусственный интеллект» скрещиваются и переплетаются проблемы, которые давно волнуют специалистов самых разных научных направлений. Психологи и программисты, философы и инженеры, лингвисты и математики, биологи и кибернетики – все они в той или иной мере соприкасаются с проблемами искусственного интеллекта и участвуют в их решении.

Интерес к моделированию рассуждений не случаен. Интеллектуальные системы создаются для того, чтобы овеществлять в технических устройствах знания и умения, которыми обладают люди, чтобы решать задачи, относимые к области творческой деятельности человека, не хуже людей.

В интеллектуальные системы, особенно в те, которые получили название экспертных систем и предназначены для помощи специалистам в решении их задач, необходимо вложить знание о том, как мы рассуждаем, когда ищем решение. И если не говорить о математике и еще нескольких науках, опирающихся на точные и формальные модели, то наши схемы рассуждений – это тот самый аппарат, с помощью которого осуществляется значительная доля творческой деятельности.

Когда специалисты в области моделирования человеческих рассуждений начали свою работу, они столкнулись с тем, что человеческие рассуждения представляют собой нечто загадочное и детально никем не изучались. Казалось бы, в логике – науке о рассуждениях – за многие века ее существования должны были накопиться горы фактов о том, как люди делают выводы на основании знаний.

Но, как выяснилось, логиков традиционно интересует лишь чрезвычайно узкий класс рассуждений, которые можно было бы назвать строгими, а остальные многочисленные формы человеческих рассуждений они не включают в свою компетенцию.

Психология мышления также весьма сдержанно относится к тому, как формируются у человека схемы рассуждений и как он ими пользуется в конкретных ситуациях. Лингвисты, которые много занимались логическими проблемами естественного языка, остались далеки от понимания того, как носитель этого языка строит на нем свои схемы принятия решений.

Цель курсовой работы ― рассмотреть модели рассуждений, их виды и цель их создания.

Актуальность темы моделирования рассуждений представляют интерес для специалистов по интеллектуальным системам и искусственному интеллекту.

Глава 1. Знания и их представление

Языки, предназначенные для описания предметных областей называются языками представления знаний. Универсальным языком представления знаний является естественный язык. Однако использование естественного языка в системах машинного представления знаний наталкивается на ряд препятствий, главным из которых является отсутствие формальной семантики естественного языка.

Системы, основанные на знаниях - это системы программного обеспечения, основными структурными элементами которых являются база знаний и механизм логических выводов. В первую очередь к ним относятся экспертные системы, способные диагностировать заболевания, оценивать потенциальные месторождения полезных ископаемых, осуществлять обработку естественного языка, распознавание речи и изображений и т.д. Экспертные системы являются первым шагом в практической реализации исследований в области ИИ. В настоящее время они уже используются в промышленности.

Экспертная система - это вычислительная система, в которую включены знания специалистов о некоторой конкретной проблемной области и которая в пределах этой области способна принимать экспертные решения. Структурные элементы, составляющие систему, выполняют следующие функции. База знаний - реализует функции представления знаний в конкретной предметной области и управление ими. Механизм логических выводов - выполняет логические выводы на основании знаний, имеющихся в базе знаний. Пользовательский интерфейс - необходим для правильной передачи ответов пользователю, иначе пользоваться системой крайне неудобно.

Модуль приобретения знаний - необходим для получения знаний от эксперта, поддержки базы знаний и дополнения ее при необходимости. Модуль ответов и объяснений - формирует заключение экспертной системы и представляет различные комментарии, прилагаемые к заключению, а также объясняет мотивы заключения.

Перечисленные структурные элементы являются наиболее характерными, хотя в реальных экспертных системах их функции могут быть соответствующим образом усилены или расширены.

Знания в базе знаний представлены в конкретной форме и организация базы знаний позволяет их легко определять, модифицировать и пополнять. Решение задач с помощью логического вывода на основе знаний хранящихся в базе знаний, реализуется автономным механизмом логического вывода. Хотя оба эти компонента системы с точки зрения ее структуры являются независимыми, они находятся в тесной связи между собой и определение модели представления знаний накладывает ограничения на выбор соответствующего механизма логических выводов. Таким образом, при проектировании экспертных систем необходимо анализировать оба указанных компонента. Чтобы манипулировать знаниями из реального мира с помощью компьютера, необходимо осуществлять их моделирование. К основным моделям представления знаний относятся:

* логические модели;
* продукционные модели;
* сетевые модели;
* фреймовые модели.

1.1 Логические модели

Логическая (предикатная) модель представления знаний основана на алгебре высказываний и предикатов, на системе аксиом этой алгебры и ее правилах вывода. Из предикатных моделей наибольшее распространение получила модель предикатов первого порядка, базирующаяся на термах (аргументах предикатов - логических констант, переменных, функций), предикатах (выражениях с логическими операциями).

Пример. Возьмем утверждение: "Инфляция в стране превышает прошлогодний уровень в 2 раза". Это можно записать в виде логической модели: r(InfNew, InfOld, n), где r(x,y) - отношение вида "x=ny", InfNew - текущая инфляция в стране, InfOld - инфляция в прошлом году. Тогда можно рассматривать истинные и ложные предикаты, например, r(InfNew, InfOld, 2)=1, r(InfNew, InfOld, 3)=0 и т.д. Очень полезные операции для логических выводов - операции импликации, эквиваленции.

Логические модели удобны для представления логических взаимосвязей между фактами, они формализованы, строги (теоретические), для их использования имеется удобный и адекватный инструментарий, например, язык логического программирования Пролог.

В основе моделей такого типа лежит понятие формальной системы. Постановка и решение любой задачи связаны с определенной предметной областью. Так, решая задачу составления расписания обработки деталей на металлорежущих станках, мы вовлекаем в предметную область такие объекты, как конкретные станки, детали, интервалы времени и общие понятия "станок", "деталь", "тип станка" и т.д.

Все предметы и события, которые составляют основу общего понимания необходимой для решения задачи информации, называются предметной областью. Мысленно предметная область представляется состоящей из реальных объектов, называемых сущностями. Сущности предметной области находятся в определенных отношениях друг к другу. Отношения между сущностями выражаются с помощью суждений. В языке (формальном или естественном) суждениям отвечают предложения.

Для представления математического знания в математической логике пользуются логическими формализмами - исчислением высказываний и исчислением предикатов. Эти формализмы имеют ясную формальную семантику и для них разработаны механизмы вывода. Поэтому исчисление предикатов было первым логическим языком, который применяли для формального описания предметных областей, связанных с решением прикладных задач.

Описания предметных областей, выполненные в логических языках, называются логическими моделями. Логические модели, построенные с применением языков логического программирования, широко применяются в базах знаний и экспертных системах.

1.2 Продукционные модели

Продукционная модель представления знаний является развитием логических моделей в направлении эффективности представления и вывода знания.

Продукция – это выражение, содержащее ядро, интерпретируемое фразой «Если А, то В», имя, сферу применения, условие применимости ядра и постусловие, представляющее собой процедуру, которую следует выполнить после успешной реализации ядра. Все части, кроме ядра, являются необязательными.

Взаимосвязанный набор продукций образует систему. Основная проблема вывода знания в системе продукций является выбор для анализа очередной продукции. Конкурирующие продукции образуют фронт.

Продукции (наряду с сетевыми моделями) являются наиболее популярными средствами представления знаний в системах ИИ. Импликация может истолковываться в обычном логическом смысле как знак логического следования B из истинного А. Возможны и другие интерпретации продукции, например А описывает некоторое условие, необходимое, чтобы можно было совершить действие B.

Если в памяти системы хранится некоторый набор продукций, то они образуют систему продукций. В системе продукций должны быть заданы специальные процедуры управления продукциями, с помощью которых происходит актуализация продукций и выполнение той или иной продукции из числа актуализированных.

В состав системы продукций входит база правил (продукций), глобальная база данных и система управления. База правил - это область памяти, которая содержит совокупность знаний в форме правил вида ЕСЛИ - ТО. Глобальная база данных - область памяти, содержащая фактические данные (факты). Система управления формирует заключения, используя базу правил и базу данных. Существуют два способа формирования заключений - прямые выводы и обратные выводы.

В прямых выводах выбирается один из элементов данных, содержащихся в базе данных, и если при сопоставлении этот элемент согласуется с левой частью правила (посылкой), то из правила выводится соответствующее заключение и помещается в базу данных или исполняется действие, определяемое правилом, и соответствующим образом изменяется содержимое базы данных. В обратных выводах процесс начинается от поставленной цели. Если эта цель согласуется с правой частью правила (заключением), то посылка правила принимается за подцель или гипотезу. Этот процесс повторяется до тех пор, пока не будет получено совпадение подцели с данными. При большом числе продукций в продукционной модели усложняется проверка непротиворечивости системы продукций, т.е. множества правил. Поэтому число продукций, с которыми работают современные системы ИИ, как правило, не превышают тысячи.

1.3 Сетевые модели

В основе моделей этого типа лежит конструкция, названная раньше семантической сетью. Семантический подход к построению систем ИИ находит применение в системах понимания естественного языка, в вопросно-ответных системах, в различных предметно - ориентированных системах.

В самом общем случае семантическая сеть представляет собой информационную модель предметной области и имеет вид графа, вершины которого соответствуют объектам предметной области, а дуги - отношениям между ними. Дуги могут быть определены разными методами, зависящими от вида представляемых знаний. Обычно дуги, используемые для представления иерархии, включают дуги типа "множество", "подмножество", "элемент". Семантические сети, используемые для описания естественных языков, используют дуги типа "агент", "объект", "реципиент". В семантических сетях существует возможность представлять знания более естественным и структурированным образом, чем в других формализмах.

1.4 Фреймовые модели

Фреймовая модель представления знаний задает остов описания класса объектов и удобна для описания структуры и характеристик однотипных объектов (процессов, событий) описываемых фреймами - специальными ячейками (шаблонами понятий) фреймовой сети (знания).

Фрейм - концентратор знаний и может быть активизирован как отдельный автономный элемент и как элемент сети. Фрейм - это модель кванта знаний (абстрактного образа, ситуации), активизация фрейма аналогична активизации этого кванта знаний - для объяснения, предсказания и т.п. Отдельные характеристики (элементы описания) объекта называются слотами фрейма. Фреймы сети могут наследовать слоты других фреймов сети.

Различают фреймы-образцы (прототипы), хранящиеся в базе знаний, и фреймы-экземпляры, создаваемые для отображения реальных ситуаций для конкретных данных.

Фреймовое представление данных достаточно универсальное. Оно позволяет отображать знания с помощью:

фрейм-структур - для обозначения объектов и понятий;

фрейм-ролей - для обозначения ролевых обязанностей;

фрейм-сценариев - для обозначения поведения;

фрейм-ситуаций - для обозначения режимов деятельности, состояний.

Пример. Фрейм-структурами являются понятия "заем", "вексель", "кредит". Фрейм-роли - "кассир", "клиент", "сервер". Фрейм-сценарии - "страхование", "банкинг", "банкротство". Фрейм-ситуации - "эволюция", "функционирование", "безработица".

Пример. Например, возьмем такое понятие, как "функция". Различные функции могут отличаться друг от друга, но существует некоторый набор формальных характеристик для описания любой функции (фрейм "Функция"): тип и допустимое множество изменений аргумента (область определения функции), тип и допустимое множество значений функции (множество значений функции), аналитическое правило связи аргумента со значением функции. Соответственно, могут быть определены фреймы "Аргумент", "Значение функции", "Закон соответствия". Далее можно определить фреймы "Тип аргумента", "Вычисление значения функции", "Операция" и др. Пример слотов для фрейма "Закон соответствия": аналитический способ задания закона; сложность вычисления (реализации).

В отличие от моделей других типов во фреймовых моделях фиксируется жесткая структура информационных единиц, называемых фреймами. Фрейм является формой представления некоторой ситуации, которую можно (или целесообразно) описывать некоторой совокупностью понятий и сущностей. В качестве идентификатора фрейму присваивается имя фрейма. Это имя должно быть единственным во всей фреймовой системе.

Фрейм имеет определенную внутреннюю структуру, состоящую из множества элементов, называемых слотами, которым также присваиваются имена. Каждый слот в свою очередь представляется определенной структурой данных. В значение слота подставляется конкретная информация, относящаяся к объекту, описываемому этим фреймом.

Значением слота может быть практически что угодно: числа, формулы, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма или других фреймов. В качестве значения слота может выступать набор слотов более низкого уровня, что позволяет реализовывать во фреймовых представлениях "принцип матрешки".

Связи между фреймами задаются значениями специального слота с именем "Связь". Часть специалистов по системам ИИ считают, что нет необходимости выделять фреймовые модели представления знаний, так как в них объединены все основные особенности моделей остальных типов.

Глава 2. Моделирование рассуждений

Представление знаний - одно из наиболее сформировавшихся направлений искусственного интеллекта. Традиционно к нему относилась разработка формальных языков и программных средств для отображения и описания так называемых когнитивных структур. Сегодня к представлению знаний причисляют также исследования по дескриптивной логике, логикам пространства и времени, онтологиям.

Пространственные логики позволяют описывать конфигурацию пространственных областей, объектов в пространстве; изучаются также семейства пространственных отношений. В последнее время эта область, из-за тесной связи с прикладными задачами, становится доминирующей в исследованиях по представлению знаний.

Объектами дескриптивной логики являются так называемые концепты, (базовые структуры для описания объектов в экспертных системах) и связанные в единое целое множества концептов (агрегированные объекты). Дескриптивная логика вырабатывает методы работы с такими сложными концептами, рассуждений об их свойствах и выводимости на них. Дескриптивная логика может быть использована, кроме того, для построения объяснительной компоненты базы знаний.[4]

Наконец, онтологические исследования посвящены способам концептуализации знаний и методологическим соображениям о разработке инструментальных средств для анализа знаний.

Различные способы представления знаний лежат в основе моделирования рассуждений, куда входят: моделирование рассуждений на основе прецедентов (case-based reasoning, CBR), аргументации или ограничений, моделирование рассуждений с неопределенностью, рассуждения о действиях и изменениях, немонотонные модели рассуждений, и другие. Остановимся на некоторых из них.

2.1 Рассуждения на основе прецедентов

Моделирование рассуждений на основе прецедентов также называют методом "ближайшего соседа" ("nearest neighbour") относится к классу методов, работа которых основывается на хранении данных в памяти для сравнения с новыми элементами. При появлении новой записи для прогнозирования находятся отклонения между этой записью и подобными наборами данных, и наиболее подобная (или ближний сосед) идентифицируется.

Например, при рассмотрении нового клиента банка, его атрибуты сравниваются со всеми существующими клиентами данного банка (доход, возраст и т.д.). Множество "ближайших соседей" потенциального клиента банка выбирается на основании ближайшего значения дохода, возраста и т.д.

При таком подходе используется термин "k-ближайший сосед" ("k-nearest neighbour"). Термин означает, что выбирается k "верхних" (ближайших) соседей для их рассмотрения в качестве множества "ближайших соседей". Поскольку не всегда удобно хранить все данные, иногда хранится только множество "типичных" случаев. В таком случае Прецедент - это описание ситуации в сочетании с подробным указанием действий, предпринимаемых в данной ситуации.

Подход, основанный на прецедентах, условно можно поделить на следующие этапы:

* сбор подробной информации о поставленной задаче;
* сопоставление этой информации с деталями прецедентов, хранящихся в базе, для выявления аналогичных случаев;
* выбор прецедента, наиболее близкого к текущей проблеме, из базы прецедентов;
* адаптация выбранного решения к текущей проблеме, если это необходимо;
* проверка корректности каждого вновь полученного решения;
* занесение детальной информации о новом прецеденте в базу прецедентов.

Таким образом, вывод, основанный на прецедентах, представляет собой такой метод анализа данных, который делает заключения относительно данной ситуации по результатам поиска аналогий, хранящихся в базе прецедентов.

Данный метод по своей сути относится к категории "обучение без учителя", т.е. является "самообучающейся" технологией, благодаря чему рабочие характеристики каждой базы прецедентов с течением времени и накоплением примеров улучшаются. Разработка баз прецедентов по конкретной предметной области происходит на естественном для человека языке, следовательно, может быть выполнена наиболее опытными сотрудниками компании - экспертами или аналитиками, работающими в данной предметной области.

Однако это не означает, что CBR-системы самостоятельно могут принимать решения. Последнее всегда остается за человеком, данный метод лишь предлагает возможные варианты решения и указывает на самый "разумный" с ее точки зрения.

Преимущества метода

* Простота использования полученных результатов.
* Решения не уникальны для конкретной ситуации, возможно их использование для других случаев.
* Целью поиска является не гарантированно верное решение, а лучшее из возможных.

Недостатки метода

* Данный метод не создает каких-либо моделей или правил, обобщающих предыдущий опыт, - в выборе решения они основываются на всем массиве доступных исторических данных, поэтому невозможно сказать, на каком основании строятся ответы.
* Существует сложность выбора меры "близости" (метрики). От этой меры главным образом зависит объем множества записей, которые нужно хранить в памяти для достижения удовлетворительной классификации или прогноза. Также существует высокая зависимость результатов классификации от выбранной метрики.
* При использовании метода возникает необходимость полного перебора обучающей выборки при распознавании, следствие этого - вычислительная трудоемкость.
* Типичные задачи данного метода - это задачи небольшой размерности по количеству классов и переменных.

Методы CBR уже применяются во множестве прикладных задач – в медицине, управлении проектами, для анализа и реорганизации среды, для разработки товаров массового спроса с учетом предпочтений разных групп потребителей, и т.д. Следует ожидать приложений методов CBR для задач интеллектуального поиска информации, электронной коммерции (предложение товаров, создание виртуальных торговых агентств), планирования поведения в динамических средах, компоновки, конструирования, синтеза программ.

2.2 Моделирование рассуждений на основе ограничений

Наиболее интересны задачи моделирования рассуждений, основанных на процедурных динамических ограничениях. Они мотивированы сложными актуальными задачами – например, планированием в реальной обстановке.

Под задачей удовлетворения ограничений понимается четверка множеств: множество переменных, множество соответствующих областей переменных, множество ограничений на переменные и множество отношений над областями. Решением проблемы удовлетворения ограничений называется набор значений переменных, удовлетворяющих ограничениям на переменные, такой, что при этом области, которым принадлежат эти значения, удовлетворяют отношениям над областями.

2.3 Немонотонные модели рассуждений

Сюда относятся исследования по логике умолчаний (default logic), по логике "отменяемых" (Defeasible) рассуждений, логике программ, теоретико - аргументационой характеризации логик с отменами, характеризации логик с отношениями предпочтения, построению эквивалентных множеств формул для логик с очерчиванием (circumscription) и некоторые другие.

Такого рода модели возникают при реализации индуктивных рассуждений, например, по примерам; связаны они также с задачами машинного обучения и некоторыми иными задачами. В частности, в задачах моделирования рассуждений на основе индукции источником первоначальных гипотез служат примеры. Если некоторая гипотеза Н возникла на основе N положительных примеров (например, экспериментального характера), то никто не может дать гарантии, что в базе данных или в поле зрения алгоритма не окажется N+1 - й пример, опровергающий гипотезу (или меняющий степень ее истинности). Это означает, что ревизии должны быть подвержены и все следствия гипотезы H.

2.4 Рассуждения о действиях и изменениях

Большая часть работ в этой области посвящена применениям ситуационного исчисления - формализма, предложенного Джоном Маккарти в 1968 году для описания действий, рассуждений о них и эффектов действий. Для преобразования плана поведения робота в исполняемую программу, достигающую с некоторой вероятностью фиксированной цели, вводится специальное логическое исчисление, основанное на ситуационной логике.

Для этой логики предложены варианты реализации на языке pGOLOG - версии языка GOLOG, содержащей средства для введения вероятностей. Активно исследуются логики действий, применение модальных логик для рассуждений о знаниях и действиях.

2.5 Рассуждения с неопределенностью

Сюда относится использование байесовского формализма в системах правил и сетевых моделях. Байесовские сети – это статистический метод обнаружения закономерностей в данных. Для этого используется первичная информация, содержащаяся либо в сетевых структурах либо в базах данных. Под сетевыми структурами понимается в этом случае множество вершин и отношений на них, задаваемое с помощью ребер. Содержательно, ребра интерпретируются как причинные связи. Всякое множество вершин Z, представляющее все пути между некоторыми двуми иными вершинами X и Y соответствует условной зависимости между этими двуми последними вершинами.

Далее задается некоторое распределение вероятностей на множестве переменных, соответствующих вершинам этого графа и полученная, но минимизированная (в некотором смысле) сеть называется байсовской сетью.

На такой сети можно использовать, так называемый байесовский вывод, т.е. для вычисления вероятностей следствий событий можно использовать (с некоторой натяжкой) формулы теории вероятностей.

Иногда рассматриваются так называемые гибридные байесовские сети, с вершинами которых связаны как дискретные, так и непрерывные переменные. Байесовские сети часто применяются для моделирования технических систем.

Глава 3. Логическое программирование

Проблемы с логической интерпретацией человеко-машинного взаимодействия имеют весьма серьезные причины, основной из которой является фундаментальное противоречие, существующее между понятием взаимодействия и декларативной семантикой логических программ. Дело в том, что программа, взаимодействующая с человеком, является частным случаем так называемых реагирующих систем, отвечающих на сообщения извне.

Характерной особенностью реагирующей системой является ее последовательный переход из одних состояний в другие под воздействием внешних событий. В то же время каждая логическая программа имеет декларативную семантику и может быть представлена в виде формулы (является статической системой).

Следует отметить, что в процессе развития логического программирования было предложено большое количество способов имитации объектов с изменяемым состоянием, основанных на использовании задержанных вычислений и параллельных стратегий исполнения логических программ.

Основные преимущества и особенности логической парадигмы программирования:

– программа, построенная на основе логического подхода, не является алгоритмом, а представляет собой запись условия задачи на языке формальной логики;

– данный подход наиболее удобен для решения логических задач;

–логическое программирование требует особого стиля мышления программиста.

Применение логических языков программирования осуществляется для описания различных утверждений, логики, рассуждений, а также для создания средств искусственного интеллекта. Известен ряд способов, позволяющих реализовать основные базовые алгоритмы на языке логического программирования.

Наиболее известный пример языка логического программирования – это PGOLOG, был разработан в середине 1970-х во Франции в рамках проекта по пониманию естественного языка.

Язык ПРОЛОГ предназначен для представления и использования знаний в различных предметных областях. Математическую основу языка составляет исчисление предикатов первого порядка (ИППП), при этом объекты предметной области, их свойства и связи представляются конъюнкцией правильно построение формул специального вида, называемых дизъюнктами Хорна. Для решения задачи получения новой информации об отношениях предметной области, формулируемой как задача доказательства теоремы, в интерпретаторе системы программирования ПРОЛОГ реализован метод резолюции.[6]

Программа на языке ПРОЛОГ состоит из утверждений (предложений, дизъюнктов Хорна), составляющих базу фактов и базу правил, к которым допустимо обращение с запросами. Запросы называются также целевыми утверждениями или просто целями.

Программирование на ПРОЛОГ состоит из следующих этапов:

1. Объявление некоторых фактов об объектах и отношениях между ними.
2. Объявление некоторых правил об объектах и отношениях между ними.
3. Формулировка вопросов об объектах и отношениях между ними.

Большинство существующих систем программирования ПРОЛОГ являются комбинированными в том смысле, что допускают как интерпретацию программ, так и их компиляцию.

3.1 Арифметика

Для описания арифметических операций в языке ПРОЛОГ используются структуры, функторами которых выступают знаки арифметических действий, а компонентами - термы, являющиеся операндами. В качестве операндов могут использоваться числа, переменные и структуры. Последние, в свою очередь, должны представлять собой арифметические выражения. С точки зрения ИППП знаки арифметических операций в таких структурах выступают в качестве функциональных букв.

Арифметические операции могут также использоваться для вычислений. Например, если имеются сведения о населении и площади некоторой страны, то можно вычислить среднюю плотность населения для этой страны. Средняя плотность населения показывает, сколь тесно было бы в данной стране, если бы ее население было равномерно распределено по всей ее территории.

Рассмотрим следующую базу данных, содержащую сведения о населении и площади некоторых стран в 1976 г. Для представления связи между страной и ее населением будет использоваться предикат нас. В наши дни население обычно характеризуется довольно большими числами. Не все версии Пролога позволяют работать с такими числами. Поэтому будем исчислять население в миллионах: нас (Х, Y) означает, что население страны X составляет примерно «Y миллионов» людей. Предикат площадь будет обозначать связь между страной и ее площадью (в миллионах квадратных километров):

нас(сша,203).

нас(индия, 548).

нас(китай,800).

нас(бразилия,108).

площадь(сша,8).

площадь(индия,3).

площадь(китай,9).

площадь(бразилия,8).

Теперь для того, чтобы найти среднюю плотность населения некоторой страны, мы должны использовать правило, гласящее, что значение плотности получается делением числа, представляющего население, на число, представляющее площадь.

Введем предикат плотность(Х, Y), где X – это страна, a Y – плотность населения в данной стране, и запишем соответствующее правило на Прологе:

плотность(X,Y):-нас(Х,Р), площадь(Х,А), Y is Р/А.

Данное правило читается следующим образом:

«Плотность населения страны X представляется числом Y, если: Население X - это Р, и Площадь X - это A, и Y вычисляется делением Р на A.»

В правиле используется оператор деления '/' введенный в предыдущем разделе. Операция деления выполняется на самом деле как целочисленное деление, сохраняющее только целую часть результата.

3.2 Запросы

Введенные элементы языка ПРОЛОГ позволяют рассмотреть более сложные типы запросов к базе фактов. В общем виде запрос (целевое утверждение) формулируется в следующем виде:

?-<структура-1>,..., <структура-N>.

Здесь каждая структура представляет собой предикат, возможно содержащий переменные. Причем областью действия переменной является всё утверждение в целом, то есть одна и та же переменная в пределах утверждения означает один и тот же объект. Символ "," (запятая) между предикатами трактуется как логическая связка И, то есть запрос необходимо рассматривать как требование на поиск в базе фактов информации, удовлетворяющей одновременно всем предикатам целевого утверждения. Предикаты, объединенные связкой И в таком запросе, называются подцелями (имея в виду весь запрос целью).

Пример. Пусть необходимо получить из базы фактов информацию о наличии строительных блоков высотой более 2 и менее 5. Тогда соответствующий диалог с интерпретатором ПРОЛОГа будет иметь следующий вид (рисунок 1):



Рисунок 1.Диалог

Получив запрос (целевое утверждение), состоящий из нескольких предикатов, интерпретатор пытается выполнить его, используя линейную по входу стратегию метода резолюции, реализуя следующую схему вычислений, поясняемую последним рассмотренным примером.

Выбирается первый в последовательности запроса предикат и делается попытка (если этот предикат не встроенный) согласовать его с базой фактов, для чего выполняется сопоставление этого предиката последовательно со всеми утверждениями базы до тех пор, пока оно не даст положительного результата. Если этого не происходит, интерпретатор выдает в качестве ответа No. В ходе согласования возможна конкретизация переменных значениями. В нашем примере на этом этапе будет выполнено согласование предиката block(F,H) с третьим утверждением базы фактов, при этом переменная F получит значение cyl, переменная Н - 5, а маркер будет установлен на утверждении 3.

Удовлетворив один предикат (подцель) запроса, интерпретатор переходит к соседнему справа (у нас это H>2). Этот предикат - встроенный, для его проверки нет необходимости обращаться к базе фактов, поскольку интерпретатор в состоянии сам установить его истинность (5>2). Далее следует проверка (попытка "согласовать") последнего предиката в запросе, которая заканчивается неудачей. Здесь включается в работу механизм бэктрекинга (возврата), который заставляет интерпретатор, передвигаясь по предикатам целевого утверждения справа налево, вновь согласовывать эти предикаты, но уже на новых утверждениях базы фактов. Если попытка пересогласовать какой-либо предикат (подцель) интерпретатору удается, то он продолжает рассмотрение подцелей от данной в обычном порядке (слева направо). В нашем примере бэктрекинг достигнет предиката block(F,H), это приведет к расконкретизации переменных F и H и поиску нового утверждения, согласующегося с ним, начиная со следующего после отмеченного маркером (то есть начиная с четвертого) факта.

Этот (четвертый) факт согласуется с подцелью, F получает значение cyl, H - 3, и это значение H удовлетворяет двум оставшимся в запросе предикатам. Таким образом, будет найден первый ответ на запрос. Ввод пользователем символа ";" инициирует механизм бэктрекинга и, следовательно, дальнейший поиск с использованием рассмотренной схемы.

3.3 Правила

Правило представляет собой дизъюнкт Хорна, содержащий один положительный литерал и несколько отрицательных, и записывается следующим образом:

<структура-0>:- <структура-1>,..., <структура-N>.

Здесь каждая структура представляет собой предикат, областью действия переменных является все правило. Предикат, стоящий слева от атома ":-", называется заголовком правила, все остальные предикаты образуют его тело. Правило может трактоваться следующим образом: предикат, являющийся заголовком правила, истинен (удовлетворен) тогда, когда истинен каждый из предикатов тела правила.

Правила в Прологе используются для описания определений, запросов к базам данных, а также обращений к другим правилам и процедурам. Пример записи правил представлен на рисунке 2.

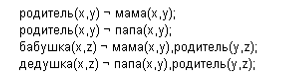


Рисунок 2.Пример записи правил

Наличие правил в программах на языке ПРОЛОГ позволяет интерпретатору находить ответ на запросы, не касающиеся непосредственно содержимого базы фактов.

В Прологе правила используются в том случае, когда необходимо сказать, что некоторый факт зависит от группы других фактов. В естественном языке для выражения правила мы можем использовать слово если. Например:

• Я пользуюсь зонтом, если идет дождь.

• Джон покупает вино, если оно дешевле, чем пиво.

В Прологе правило состоит из заголовка и тела правила. Заголовок и тело соединяются с помощью символа:-, который состоит из двоеточия: и тире -. Символ ':-' читается если.

Заголовок правила описывает факт, для определения которого предназначено это правило. Тело правила описывает конъюнкцию целей, которые должны быть последовательно согласованы с базой данных, для того чтобы заголовок правила был истинным.

Заключение

Анализируя историю ИИ, можно выделить такое обширное направление как моделирование рассуждений. Долгие годы развитие этой науки двигалось именно по этому пути, и теперь это одна из самых развитых областей в современном ИИ. Моделирование рассуждений подразумевает создание символьных систем, на входе которых поставлена некая задача, а на выходе требуется её решение.

В настоящее время разработано большое количество различных методов, подходов и путей к моделированию рассуждений.

Чтобы манипулировать знаниями из реального мира с помощью компьютера, необходимо осуществлять их моделирование.

Рассмотрев данную темы, мы пришли к выводу, что моделирование рассуждений способствует облегчению труда при создании программы на основе знаний. Так как при решение какой-либо проблемы мы ищем различные пути решения, то есть мы рассуждаем. Но бывают случаи, что задача сложная и представляет собой большой труд.

Список использованной литературы

1. Андрейчиков, А.В. Интеллектуальные информационные системы: учебник / А. В. Андрейчиков, О.Н. Андрейчикова. – М.: Финансы и статистика, 2007. – 250 с.
2. Астахова, И.С. Системы искусственного интеллекта. Практический курс: учеб. пособие / И.С. Астахова, А.С. Потапов, В.А. Чулюков. – М.: Бином, Лаборатория знаний, 2008. – 276 с.
3. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии: Учебное пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. – 304 с.
4. Гаврилова, Т.А. Базы знаний интеллектуальных систем: учебник / Т.А. Гаврилова. – СПб.: Питер, 2008.
5. Гаскаров, Д.В. Интеллектуальные информационные системы: учебник / Д.В. Гаскаров. – М.: Высшая школа, 2008.
6. Девятков, В.В. Системы искусственного интеллекта: учеб. пособие для студентов вузов / В.В. Девятков. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2010. – 254 с.
7. Золотов С.И. Интеллектуальные информационные системы: учебное пособие / С.И. Золотов – Воронеж: Научная книга, 2007. – 140 с.
8. Избачков, Ю.С. Информационные системы: учеб. пособие для ВУЗов / Ю.С. Избачков, В. Н. Петров. – СПб.: Питер, 2008.
9. Люгер Джордж Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2008. – 864 с.
10. Пескова, С.А. Сети и телекоммуникации: учебное пособие для ВУЗов России / С.А. Пескова, А.Н. Волков, А.В. Кузин. – М.: Академия, 2007.
11. Путькина, Л.В. Интеллектуальные информационные системы: учебное пособие / Л.В. Путькина, Т.Г. Пискунова. – СПб.: СПбГУП, 2008.
12. Рассел С. Искусственный интеллект: современный подход. – М.: Издательский дом «Вильямс», 2006.– 258 с.
13. Тельнов, Ю.Ф. Интеллектуальные информационные системы в экономике: учеб. пособие / Ю. Ф. Тельнов. – М.: Синтег, 2008.
14. Ясницкий Л.Н. Введение в искусственный интеллект: учебное пособие / Л.Н. Ясницкий. – М.: Академия, 2008.