

ЛЕКЦИЯ № 3: ОСНОВЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

В данной лекции рассматриваются:

- *понятие искусственного интеллекта (ИИ);*
- *классификация ИИ;*
- *направления развития ИИ;*
- *представление знаний и работа с ними;*
- *языки интеллектуальных систем.*

Цель лекции: познакомить обучающихся с основами искусственного интеллекта, основными понятиями.

Понятие искусственного интеллекта и классификация его основных направлений

Искусственный интеллект (ИИ) – это научная дисциплина, возникшая в 50-х гг. 20-го века на стыке кибернетики, лингвистики, психологии и программирования. Ее целью было стремление переложить на машину задачи, ранее традиционно решаемые людьми.

Существует достаточно много направлений использования искусственного интеллекта применительно к вычислительной технике:



С самого начала исследования в области ИИ пошли по 2-м направлениям

Направления развития искусственного интеллекта

Первое – попытки с помощью моделирования отдельных нейронов и составленных из них схем воспроизвести психофизиологическую деятельность человеческого мозга с целью создания человеческого разума. Это направление называют **бионическим**. С успе-

хами в микроэлектронике (нейрокомпьютеры) в настоящее время, это направление приобретает второе дыхание, хотя в данной сфере деятельности получено пока еще очень мало ценных результатов.

Функциональный подход – состоит в воспроизведении на ЭВМ алгоритмов, определяющих те или иные функции человеческого интеллекта. Среди них огромный интерес представляют задачи классификации, распознавания образов, логического вывода, обучения, целеполагания и принятия решений, общения на естественных языках.

В дальнейшем мы будем рассматривать только второе (функциональное) направление.

Разработка интеллектуальных программ существенно отличается от обычного программирования и ведется путем построения **системы искусственного интеллекта (СИИ)**. Если обычная программа может быть представлена в следующем виде:

ПРОГРАММА=АЛГОРИТМ+ДАННЫЕ

То система искусственного интеллекта выглядит по-другому:

СИИ=ЗНАНИЯ+СТРАТЕГИЯ ОБРАБОТКИ ЗНАНИЙ

Основным отличительным признаком СИИ является работа со знаниями. Если для обычных программ проблема представления данных алгоритма определяется на уровне языка программирования, то для СИИ представление знаний выливается в комплексную проблему: что такое знания, какие знания хранить в системе в виде базы знаний и в каком виде, как их использовать, пополнять и т.д.

Представление знаний и работа с ними

Термин «знания» в применении к информации, представимой в ЭВМ, появился не слишком давно. На более раннем этапе в ходу были лишь термины «программа» и «данные». Что же побудило специалистов ввести новый термин? Можно указать по крайней мере четыре причины, которые позволяют говорить не о данных, а о знаниях, используемых в ЭВМ.

1. **Внутренняя интерпретируемость**, т.е. вместе с информацией в базе знаний представлены информационные структуры, позволяющие не только хранить знания, но и использовать их.

Когда некоторые сведения вводятся в память ЭВМ, то каждая их порция для нее представляется в виде машинного слова, имеющего вид двоичного кода определенной длины. Что скрывается за кодом, узнать невозможно. Это может быть число, некоторый символ, закодированное слово и т.п. Для ЭВМ все эти сведения одинаковы и однотипны. Она все их интерпретирует как двоичные числа, над которыми совершаются те или иные операции. Таким образом, ЭВМ имеет аппаратно встроенную в нее интерпретацию, и изменить ее она не может. С этой точки зрения ЭВМ «однотипно» исполняет все реализуемые на ней программы. Если в некоторый момент времени остановить машину и проанализировать, что она делала, то вряд ли можно угадать, сочиняла ли она музыку или решала дифференциальное уравнение, ибо внешне она всегда делает одно и то же: складывает и умножает, делит и вычитает, сравнивает и ищет команду в программе, которую она должна выполнить следующей, и т.п. Истинная интерпретация решаемой задачи и тех данных, которые для этого привлекаются, остается в голове у программиста. Именно он заботится о том, чтобы программа выбирала нужные коды и делала с ними соответ-

ствующие операции, которые на уровне программиста могут интерпретироваться совсем не так, как на уровне прямой их интерпретации в машине.

Короче говоря, машина «не знает», что она делает на самом деле. Первый шаг на пути к знаниям в том и заключается, чтобы сообщить ей эти сведения, дать ей возможность интерпретировать данные и свои действия с ними на содержательном уровне.

Как это можно сделать, поясним на следующем примере. Пусть на ЭВМ решается задача, связанная с учетом наличия деталей на складе некоторого автохозяйства. Пусть в памяти машины хранится информация в виде таблицы (табл. 1).

Наименование единицы	Номер секции	Количество	Модель
Картер	54	200	ВАЗ 2101
Крышка картера	54	300	ВАЗ 2101
Крышка картера	7	52	ВАЗ 2106
Распределительный вал	11	1	ВАЗ 2103
Стекло переднее	11	42	ВАЗ 2101,
И т.д.			21011

Предположим *сначала*, что *в память введена лишь та часть таблицы, которая расположена снизу и справа от жирных линий*. , Данные, входящие в нее, как-то расположены в поле памяти ЭВМ. *Их использование возможно только таким образом, как это предусматривает программист в своих программах. Только программы могут найти нужные им данные и произвести с ними необходимые операции.* Интерпретация элементов введенного в машину массива оставалась вне ее.

Теперь *введем в память всю таблицу*. В этой таблице как бы *две части*. Одна из них расположена выше горизонтальной линии. *Это шапка таблицы*. В ней перечислены так называемые атрибуты. В нашем примере их четыре. Каждый из них является именем соответствующего столбца таблицы. *Вторая часть таблицы состоит из ее строк, каждая из которых содержит четыре элемента* (в правом нижнем углу таблицы тоже один элемент, но составной). Эти строки можно рассматривать как отношения, заданные на множестве элементов, находящихся справа и снизу от жирных линий. Тогда элементы левого столбца таблицы можно рассматривать как имена отношений, и *каждой строке таблицы соответствует свое имя*. Атрибуты и имена отношений дают возможность ЭВМ интерпретировать введенную в нее информацию на содержательном уровне. Теперь она вполне поймет вводимые в нее запросы такого типа, как «Сколько деталей хранится в секции 54?», «Перечислите все запасные части, которые хранятся на складе», «Имеются ли на складе распределительные валы для модели ВАЗ-2106?» и т.п.

Описанный способ внутренней интерпретации находит широкое применение при хранении информации в современных и разрабатываемых ЭВМ. Построенные на его основе базы называются реляционными. Но он не единственно возможный. Чуть далее мы рассмотрим другие способы организации внутренней интерпретации.

2. Структурированность, т.е. декомпозиция сложных объектов на более простые и установление связей между объектами.

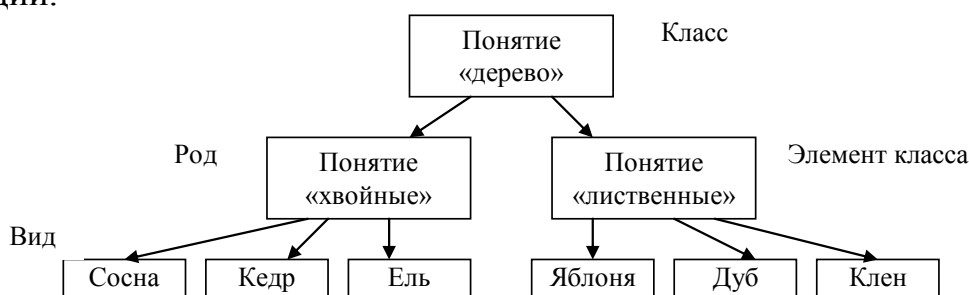
Одной из фундаментальных особенностей человеческого познания окружающего мира является его способность к декомпозиции наблюдаемых объектов, умению выделять в них отдельные элементы и связи между ними. Это позволяет человеку воспринимать любой объект как некоторую структуру и представлять сложные объекты в виде совокупности более простых и познавать свойства нового объекта через свойства тех, что

входит в его структуру. Это позволяет человеку устанавливать связи между объектами. Связи типа «часть – целое», «класс – элемент», «род – вид» и т.п. Такие связи образуют в нашей памяти иерархии понятий и позволяют вводить новые понятия, объединяющие в себе множества понятий, соответствующих реальным объектам внешнего мира. Так возникают понятия, например, типа «дерево», «млекопитающие», «мебель».

Знания, представимые в машине, должны обладать способностью к образованию иерархий, введению новых обобщенных понятий и декомпозиции понятий на составляющие их понятия и отношения между ними.

Сделать это можно через– явное указание хранящихся в базе знаний тех ее единиц, которые являются для них родовыми, и тех единиц, которые являются для них видовыми.

Важной особенностью такой организации является возможность однократной записи информации.

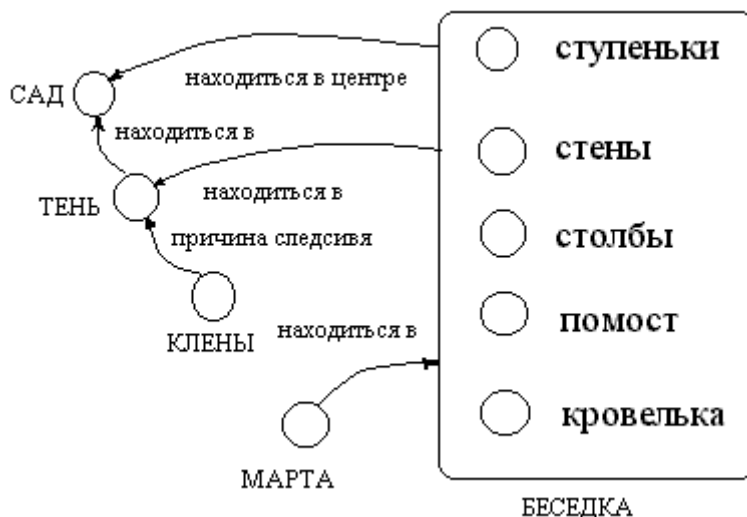


Например, вся информация, общая для всех деревьев, записана только один раз в единице информации, названной «дерево», а вся информация, общая для хвойных деревьев записана также только один раз в единице «хвойные» и не повторяется в единицах «сосна», «кедр» и «ель». В этих единицах информации содержатся сведения касающиеся непосредственно данного вида дерева, например, виды, место произрастания, целебные свойства и т.д. При необходимости получения на данном уровне детализации всей информации, относящейся к выбранной информационной единице, нужно пройти от нее вверх до самой последней в иерархии родовой единицы и собрать на этом пути всю недостающую информацию.

3. **Связанность**, т.е. свойство отображать закономерности фактов, процессов, явлений и причинно-следственные отношения между ними.

Знания, которые мы храним, связаны друг с другом не только родовыми и видовыми связями, рассмотренными нами только что. Имеются и другие связи. Среди них особое значение для нас имеют те, которые устанавливают закономерности между отдельными единицами информации (фактами, процессами, явлениями), а также определяют структуру той ситуации, в рамках которой наблюдаются сущности, которым соответствуют эти информационные единицы.

Поясним это на примере описания такой ситуации: «Среди сада в тени развесистых кленов, стояла старенькая серенькая беседка – три ступеньки вверх, обомшелый помост, низенькие стены, шесть точечных пуза-



тых столбов и шестискатная кровелька. Марта сидела в беседке ...» (Сологуб Ф. Мелкий бес. М.-Л., Академия, 1933).

Попробуем выделить в этой ситуации основные единицы информации и установить между ними связи. Во-первых, здесь имеется комплекс единиц, связанных между собой родовой единицей «беседка». В отношении часть – целое к ней находятся «ступеньки», «помост», «стены», «столбцы» и «кровелька». Кроме этих единиц, можно выделить еще следующие: «сад», «тень», «клёны», «Марта». Эти единицы связаны между собой и с единицей «беседка» связями иного типа, чем родо-видовая связь. Эти связи носят ситуативный характер. Например, между «Мартой» и «беседкой» имеется пространственная связь типа «находиться в», а между «клёнами» и «тенью» – связь типа «причина – следствие». Полная структура взаимосвязей, соответствующих описанной в тексте ситуации, показана на рис. 3.

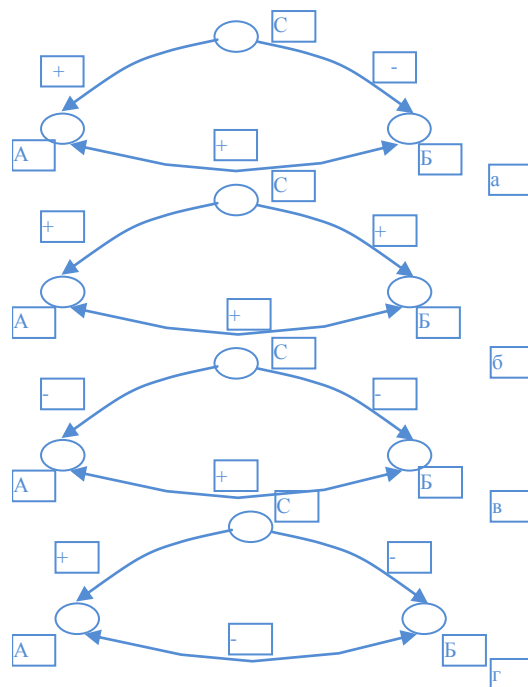
По умолчанию между всеми вершинами в этой сети имеется также отношение «быть одновременно». Родо-видовые связи в структуре понятия «беседка» также не показаны для большей наглядности рисунка. Они подобны тем, которые мы проиллюстрировали на рис. 2.

4. **Активность**, т.е. знания, обеспечивают целенаправленное использование информации.

Четвертой важной характеристикой того, что можно считать настоящими знаниями, является их активность. Исторически сложилось так, что при работе с вычислительными машинами **активными являлись программы**, т.е. процедуры, а **данные в ЭВМ оказались пассивными**. Они просто хранились в памяти и ждали, когда их извлечет оттуда нуждающаяся в них программа. **Если же обратиться к человеку, оперирующему со знаниями, то активными у него оказываются именно знания. Именно они формируют цели его деятельности** даже тогда, когда исходная мотивация его поведения лежит в глубинах бессознательного. **Активизированные цели вызывают реализацию тех процедур, которые могут оказаться полезными для достижения поставленных целей.** Зачастую сами цели являются следствием того состояния, в котором находятся знания человека об интересующем его процессе или явлении. **Неполнота знаний заставляет человека искать пути их пополнения, а противоречивость в знаниях – стремление к преодолению ее.** При создании интеллектуальных систем необходимо обеспечить воспроизведение этого важнейшего свойства «базы знаний» человека.

Как и в предыдущих случаях, имеется много способов имитации этой характеристики знаний в искусственных системах. Для примера рассмотрим один из них. На рис. 4, показаны три информационные единицы, хранящиеся в базе знаний.

Они связаны между собой особыми оценочными связями. Вершина, соответствующая самой системе (вершина С), имеет положительную оценку пользователя А. . Относительно пользователя Б у системы мнение иное, оно носит негативный характер.



Пусть в результате опроса система узнала, что пользователи А и Б высокого мнения друг и друге. Этой информации, полученной системой, соответствует положительная связь между пользователями А и Б на рис. 4 а.

Если искусственная система ведет себя подобно человеку, то структура в виде треугольника, которая показана на рис. 4 а, должна активизировать систему, поскольку подобная структура характеризует диссонанс в ее знаниях. А всякий диссонанс требует своего разрешения. Как и у человека, у системы есть четыре возможности преобразования имеющегося диссонанса в консонанс.

Первая заключается в изменении своего мнения о пользователе Б. Возможно, система не совсем правильно оценила результаты взаимодействия с ним, и необходимо собрать о пользователе Б дополнительную информацию. Для этого следует возбудить соответствующие программы ее сбора. Если после их работы система изменит свое мнение о пользователе Б, то содержимое той зоны ее базы знаний, которым мы интересуемся, станет таким, как это показано на рис. 11 б, Теперь выполнен принцип: «друзья моих друзей – мои друзья», и система из состояния диссонанса приходит в состояние, которое ее вполне устраивает.

Вторая возможность состоит в изменении своего мнения о пользователе А. Ведь система могла и ошибиться, приписав ему положительную характеристику. Снова включаются процедуры пополнения знаний о пользователях. Теперь адресатом их является пользователь А. Если после дополнительной проверки зона базы знаний становится такой, как это показано на рис. 4 в, то диссонанс опять ликвидирован. Понятно, что А и Б хорошо относятся друг к другу, но оба они с точки зрения системы не слишком хороши.

Третья возможность – возбудить процедуры в пользователях А и Б по проверке отношений между собой. Например, сообщить этим пользователям (или одному из них, например А, к которому система хорошо относится) некоторую дополнительную, располагающую системой информацию о них. Если после этого пользователи (по крайней мере А) изменят мнение друг о друге и треугольник примет вид, показанный на рис. 4 г, то система вновь будет удовлетворена. По ее мнению, «справедливость восторжествовала» и диссонанс ликвидирован.

Если же все три попытки не увенчались успехом – треугольник остался таким как он показан на рис. 4 а, - то у системы остается в запасе единственная процедура – вытеснение диссонансного знания из активного поля внимания. Возможно, что со временем появится новая информация, пока недоступная системе, которая и позволит ликвидировать этот неприятный фрагмент в знаниях системы.

Перечисленные особенности должны быть отражены в тех языках, которые используются для их представления в интеллектуальных системах. Сейчас известны **четыре вида таких языков: логические, реляционные, фреймовые и продукционные**. Рассмотрим кратко их особенности и способы применения.

Языки интеллектуальных систем

1. Языки логического типа опираются на исчисления, заимствованные из логики. Чаще всего – на исчисления предикатов. Каждый факт в базе знаний описывается в виде некоторого набора предикатов.

Если, например $P1$ есть предикат «хорошо относиться», то $P1(C, A)$ может интерпретироваться как утверждение: «Система хорошо относится к пользователю A ». Сложные структуры из фактов задаются формулировками, состоящими из предикатов и операций, принятых в этом исчислении. Например, формула $P1(C, A) \& \overline{P1}(C, B) \& P1(A, B)$ описывает ситуацию, показанную на рис. 4 а. В этой формуле черточка над символом предиката означает отрицание соответствующего утверждения, а значок $\&$ (операция конъюнкции) означает одновременность того, что все три предиката, входящих в эту формулу, являются истинными. Закономерности, характерные для проблемной области, в которой работает система, также задаются формулами исчисления предикатов. Для этого используется операция импликации, обозначаемая стрелкой \rightarrow . Запись: $(P1(C, X) \& P1(X, Y)) \rightarrow P1(C, Y)$ может характеризовать принцип типа: «Друзья моих друзей – мои друзья».

Достоинством языков логического типа является существование мощных процедур логического вывода, обеспечивающих поиск вывода в большинстве случаев. При этом довольно просто решается важная для баз знаний задача пополнения текстовых описаний, поступающих в базу знаний после работы системы общения. Т.е. сообщения, вводимые в систему, как правило, являются неполными. Поэтому, одной из основных задач является пополнение сообщений.

Недостатком, присущим всем языкам логического типа, является их плохая наглядность для человека. Весьма нелегко описывать проблемную область в виде набора аксиом, отражающих фундаментальные факты и закономерности, характерные для нее. Это приводит к многочисленным неточностям и неполноте описания, делая задачу заполнения базы знаний чрезвычайно трудоемкой.

Этим недостатком не обладают языки реляционного типа.

2. Языки реляционного типа

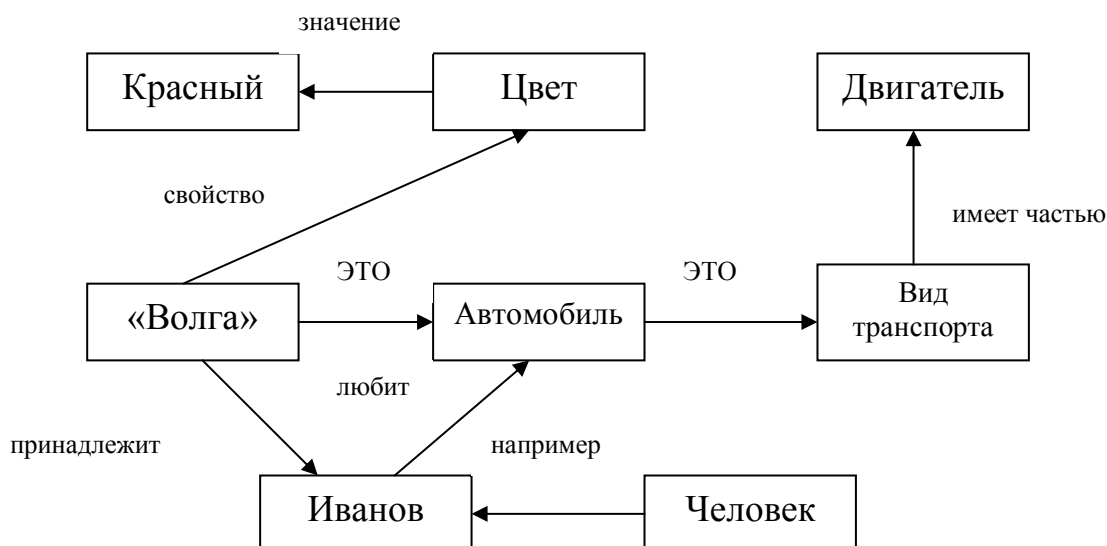
Простейшим видом реляционных языков являются табличные языки.

Язык таблиц привычен человеку. Таблицы легко обозримы, структура их прозрачна, а данные сгруппированы весьма компактно. Но табличная форма задания отношений удобна далеко не всегда.

Поэтому в системах искусственного интеллекта получили распространение реляционные языки с более широкими возможностями. Их обычно **называют языками семантических сетей**.

Семантическая сеть — это ориентированный граф, вершины которого — понятия, а дуги — отношения между ними. Понятиями обычно выступают абстрактные или конкретные объекты, а дугами — отношения между ними.

Например И Т.Д.. Пример изображен на рис. 6. В качестве вершин тут выступают понятия «человек», «т.Иванов», «Волга», «автомобиль», «вид транспорта» и «двигатель»; в качестве дуг—отношения: «это (IS—A, AKO)», «имеет частью (has part)», «принадлежит», «любит».



Характерной особенностью семантических сетей является наличие трех важнейших типов отношений: ТИПЫ ОТНОШЕНИЙ

- класс — элемент класса («это»);
- свойство — значение («цвет — красный»);
- пример элемента класса («например»).

Основным преимуществом этой модели является то, что она более других соответствует современным представлениям об организации долговременной памяти человека. Термин семантическая означает «смысловая», а сама семантика — это наука, устанавливающая отношения между символами и объектами, которые они обозначают, т. е. наука, определяющая смысл знаков. Недостатком этой модели является сложность организации процедуры поиска вывода на семантической сети.

3. Языки фреймового типа

Под фреймом понимается абстрактный образ или ситуация. формализованная модель для отображения образа.

В психологии и философии известно понятие абстрактного образа. Например, произнесение вслух слова «комната» порождает у слушающих образ комнаты: жилое помещение с четырьмя стенами, полом, потолком, окнами и дверью площадь» 10—20 м²). Из этого описания ничего нельзя убрать (например, убрав окна, мы получим уже чулан, а не комнату), но в нем есть «дырки» или слоты — незаполненные значения некоторых важных атрибутов — количество окон, цвет стен, высота потолка, покрытие пола и др.

Структура фрейма может быть представлена так:

Имя фрейма, (имя 1-го слота: значение 1-го слота),
 (имя 2-го слота: значение 2-го слота),
 (имя n-го слота: значение n-го слота)

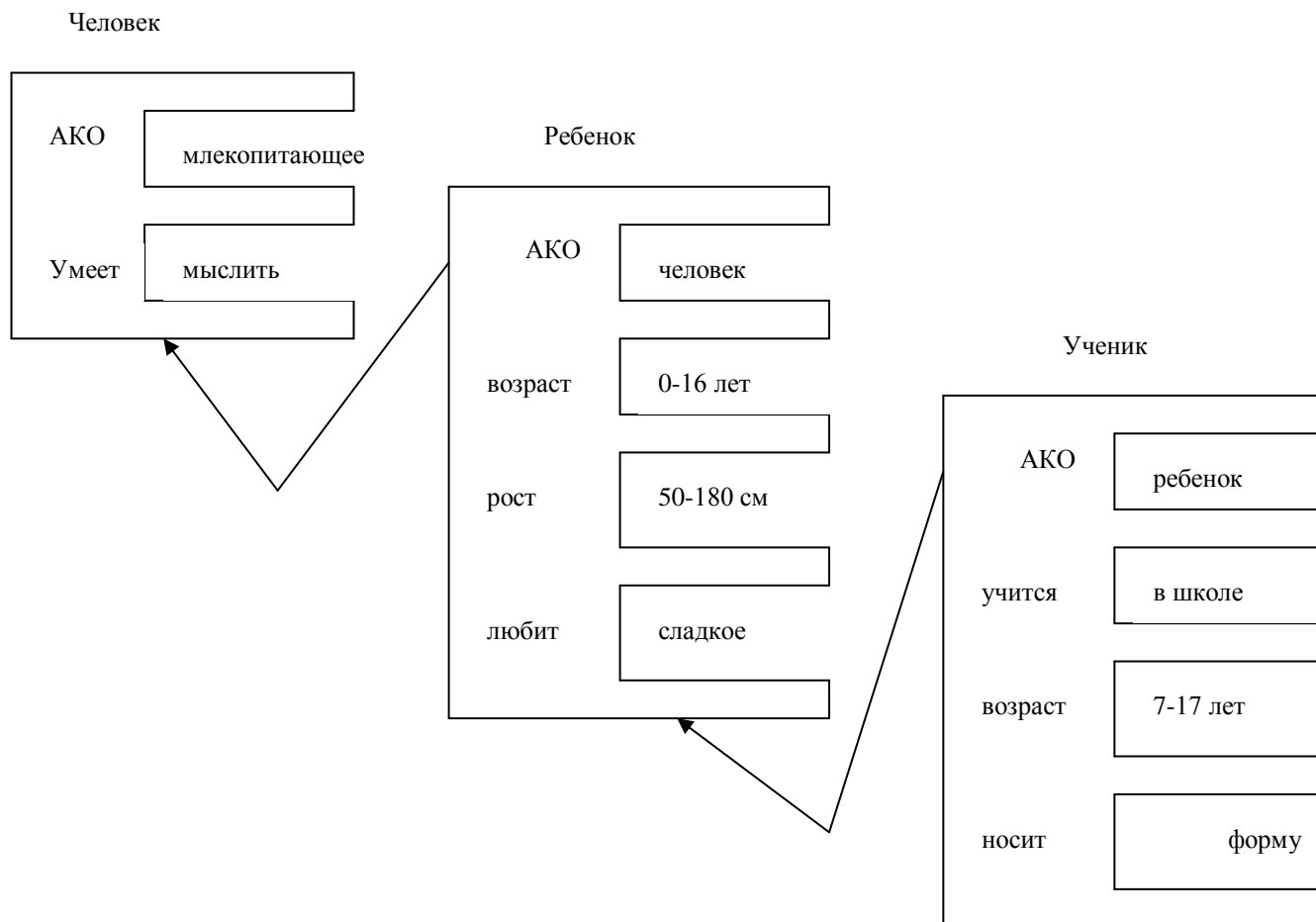
В качестве примера рассмотрим фрейм «квартира»:

квартира (<столовая> <20 кв.м>)
 (<стол> <обеденный, раскладной>)
 (<стулья> <6, мягкие>...).

Важнейшим свойством теории фреймов является заимствованное из теории семантических сетей наследование свойств.

Основным преимуществом фреймов как модели представления знаний является то, что она отражает концептуальную основу организации памяти человека, а также ее гибкость и наглядность

Например, в сети фреймов на рис. 7 понятие ученик наследует свойства фреймов «ребенок» и «человек», которые находятся на более высоком уровне иерархии. Так, на вопрос «любят ли ученики сладкое?» следует ответ «да», так как этим свойством обладают все дети, что указано во фрейме «ребенок». Наследование свойств может быть частичным — возраст для учеников не наследуется из фрейма «ребенок», так как указан явно в своем собственном, фрейме. ...



4. Языки продукционного типа

В заключение этого раздела укажем на последний класс языков представления знаний, играющий большую роль в системах искусственного интеллекта. В отличие от логических и реляционных языков, которые в прямой форме не представляют информацию о процедурах и условиях их применения, этот язык, называемый языком продукций, прямо ориентирован на представление знаний такого рода. Основным элементом этого языка является конструкция вида:

(i), P, P; A → B; (j), Q.

Словесно она может быть интерпретирована следующим текстом: «Если выполнено условие P и имеет место A, то надо сделать B и изменить условие продукции с номером j на условие Q». В нашей записи (i) есть номер написанной продукции.

В качестве примера рассмотрим формулу, задающую условие того, что у больного нет четырех перечисленных ниже заболеваний:

$$P = \bar{P}_1 \vee \bar{P}_2 \vee \bar{P}_3 \vee \bar{P}_4 \vee ,$$

где P_1 — тиреотоксикоз, P_2 — сахарный диабет, P_3 — декомпенсированная форма аддисоновой болезни, P_4 — тяжелые формы коронарной недостаточности. Пусть A — дизбактериоз, B — прием тиреоидина 0,3 г, Q — увеличить значение q на 0,3. Предикат $P_5(q>1)$ оценивает допустимость очередного приема тиреоидина, так как суточная норма для взрослого не должна превышать 1 г. Вид продукции:

(i), P , P_5 ; $A \rightarrow B$; (i), Q .

Смысл этой продукции следующий «Если у больного гипотиреоз, нет противопоказаний к приему тиреоидина, его суточная норма еще не достигнута, то необходимо принять 0,3 г тиреоидина и учесть этот факт в суточной норме». После истечения календарных суток какая-то другая продукция, следящая за течением времени, возвращает значение q к нулю.

Энтузиасты продукционного подхода к представлению знаний считают, что если в качестве значений слотов во фреймах допустить использование продукций, то такой язык окажется наиболее эффективным и вытеснит из систем искусственного интеллекта другие способы представления знаний.