1. Гносеология и семантические основы знаний  
   Гносеология – это отрасль философии или философское учение о знаниях. Важнейшая цель этого учения – найти ответ на вопрос: В чём заключается отличие между истинными (достоверными знаниями) и ложными (недостоверными знаниями). Гносеология – теория познания, а сама теория познания –это результат обобщения всей истории познавания мира, т. е. она ищет общие закономерности познания в любых сферах человеческой деятельности. Всё разнообразие познавательной деятельности сводится к трём видам: 1) Познавательно-эмпирическая (метод научного тыка), 2) Художественно-эстетический (основан на инстинктах человека), 3) Научно-теоретический. Высший формой научного знания является теория. При изучении используются определенные методы, поэтому это привело к созданию специального учения о методах научного познания. Вот этим учением и является методология
2. Методология познания. Уровни методологии

|  |  |  |
| --- | --- | --- |
| Философский | Философская теория познания, представление о научном познании | Философские науки |
| Общенаучный | Синтез частнонаучных и философских знаний | Методологические науки |
| Частно научный метод | Синтез общенаучной методологии и системы знаний соответствующей науки | Методология психологии |
| Конкретные методы и методики | Методы представления конкретной науки | Дисциплины, которые разрабатывают конкретные методы и методики |

Метод – способ реализации, а методика – это план и инструкция. Способы получения знаний по В. С. Соловьёву делятся на три: 1) Эвристический или научный, 2) Радикальный или философский, 3) Мистический способ (основанный на вере, интуиции или творчестве).

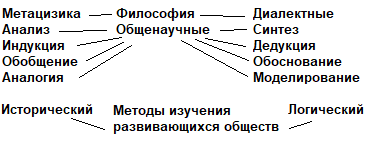
1. Метод Декарта. Методологическая гносеология: 1) Эвристическая, которая содействует появлению новых знаний, 2) Координирующая, которая занимается объединением некоторых частей, 3) Логико-гносеологическая, которая занимается обоснованием понятий и теоретических структур научного знания. Часто применяется метод Декарта. Этот метод, метод достижения истины, состоит из 4 шагов: 1) Принимаем за истину только то, что представляется уму столь отчетливо, что не может дать повод для сомнения, 2) Делить каждую из трудностей на столько частей, сколько потребуется, чтобы точно их разрешить, 3) Располагать свои мысли в строгом порядке начиная с предметов событий простейших и восходя постепенно до наиболее сложных, 4) Делать характеристики столь полными и образными сколько потребуется, чтобы быть уверенным, что ничего не пропущено.
2. Процесс познания

Есть два подхода: 1) Классический (есть объект, который не я, и есть субъект познания, который я), 2) Научный. Схема научного метода:



В результате познания мы получаем объективную информацию о явлении, объекте. Все получаемые знания должны соответствовать следующим критериям: 1) Полнота исходных оснований, 2) Выводимость знаний на основе изучения, 3) Целостность построения знаний.

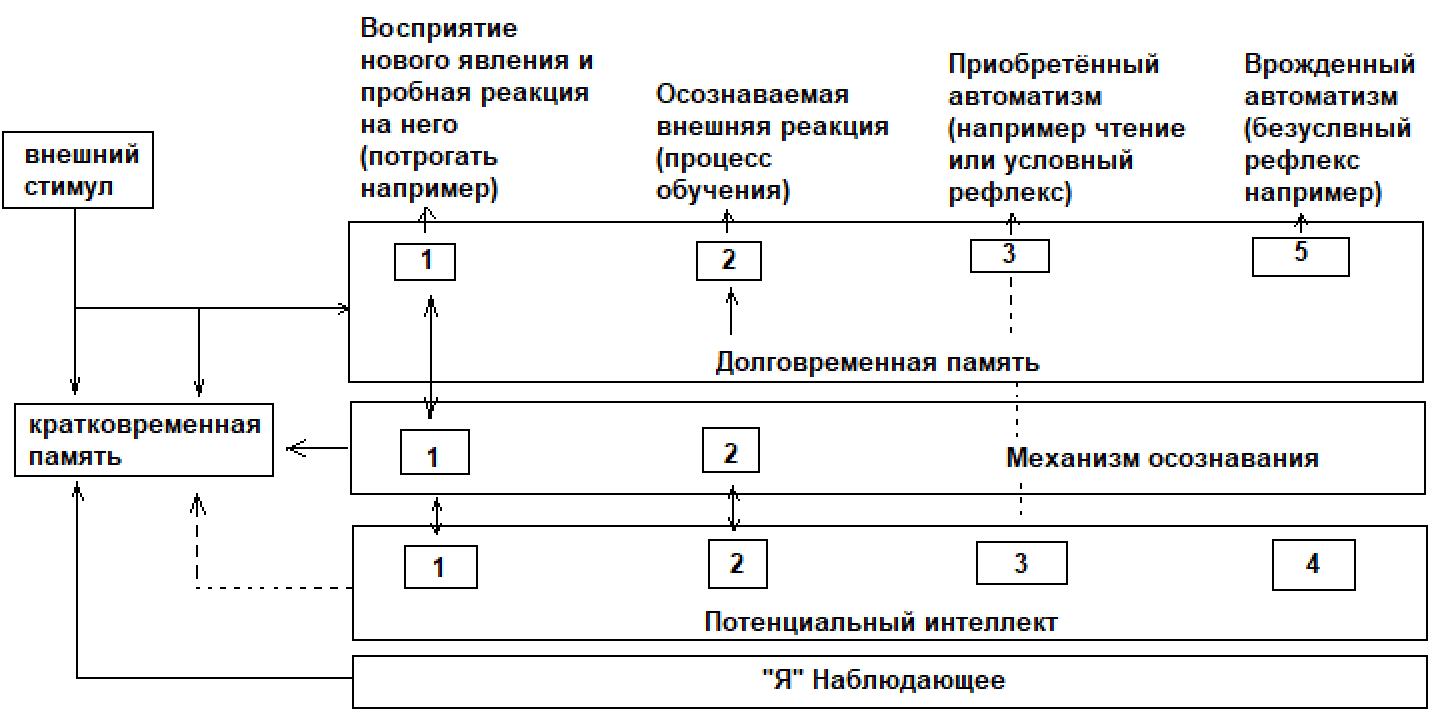
1. Общая классификация методов познания



1. Сущность интеллекта и влияние знаний на интеллект искусственной системы  
   Интеллект – совокупность всех познавательных функций индивида от ощущения и восприятия до мышления и воображения. Психологи отмечают 3 основных функций интеллекта: 1) Способность к обучению, 2) Оперирование символами, 3) Способность к активному овладению закономерностями окружающей среды.

Интеллектуальные способности:

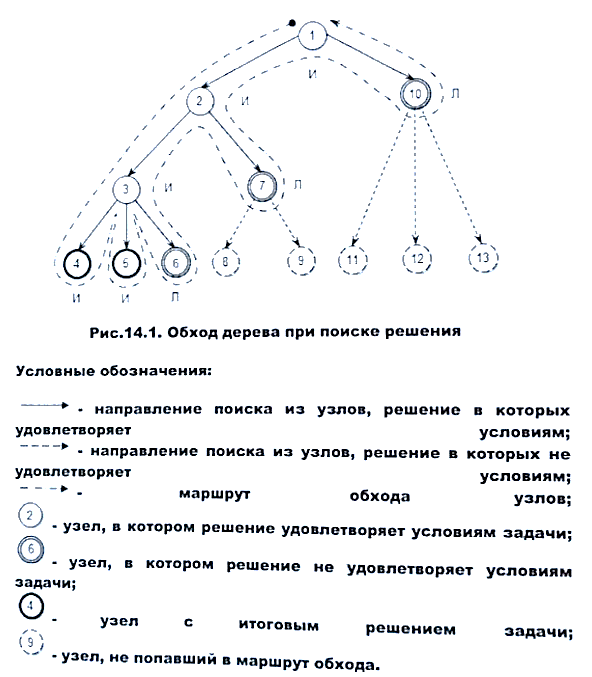
|  |  |  |
| --- | --- | --- |
| **Конвергентные способности** | **Креативные (дивергентные способности)** | **Обучаемость** |
| Репродуктивные или первичные способности | Фантазия и воображение | Способность к освоению знаний |
| Аналитические, формально-логические способности | Синтетические способности | Уровень мотивирования |
| Процессуальные способности интеллекта | Оригинальность мышления | Способность к переносу трансформации |
|  | Восприимчивость к новизне | Степень повышения интеллекта |
|  | Интуиция, развитость подсознания |  |
|  | Критичность |  |

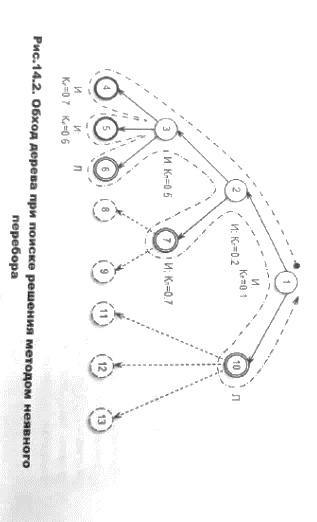
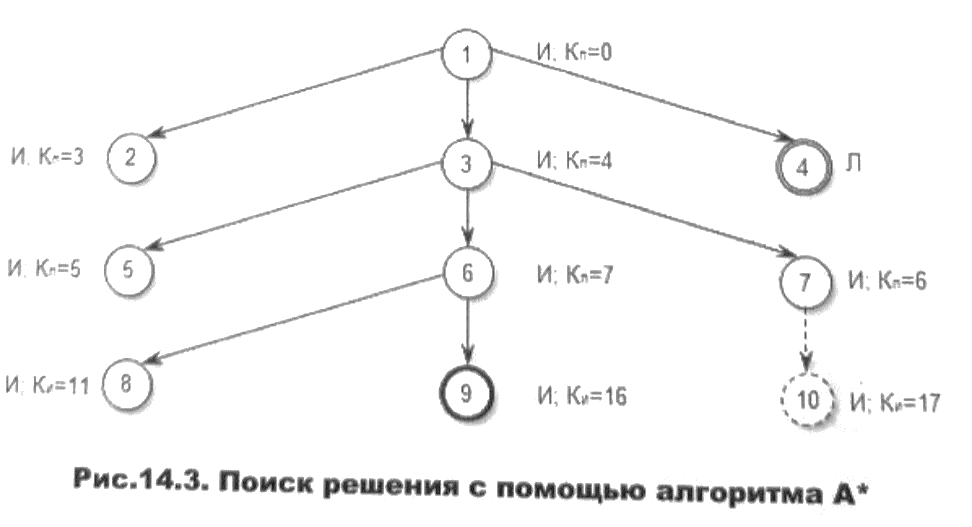
1.6.2 Модель интеллектуальной (познавательной) деятельности мозга человека.   


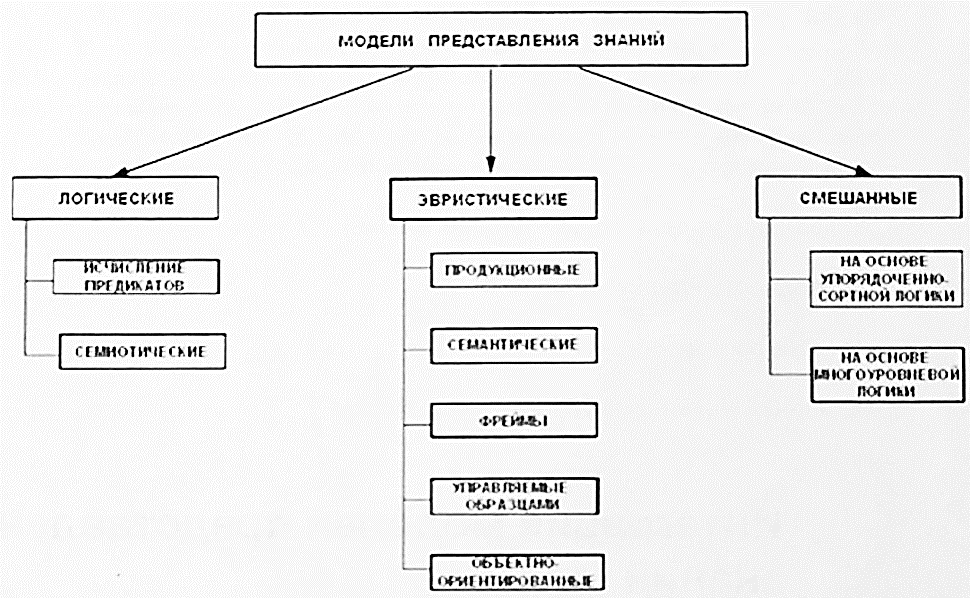
1 – первое знакомство с объектом, 2 – процесс обучения, 3 – обучение до автоматизма, 4 – невостребованная познавательная модель, 5 – врожденные сведенья.  
Всегда есть внешний стимул, он ещё называется внешним раздражителем. Это воздействие сразу фиксируется у наблюдателя в кратковременной памяти. Любой человек обладает потенциальным интеллектом, уровень знаний, умение анализировать, собирать информацию. Из кратковременной памяти раздражающее воздействие попадает в механизм осознавания, происходит восприятие нового явления и осознание его. Подключается интеллект и информация, которая хранится в долговременной памяти. Если там ничего нет о данном явлении, то мы начинаем процесс обучения. Наш потенциальный интеллект собирает информацию об объекте анализирует эту информацию, делает выводы, и результаты этого исследования уже заносятся в долговременную память. Невостребованная познавательная модель. Так формируется новое знание.   
1.6.3 Искусственный интеллект – это наука и разработка интеллектуальных машин и систем интеллектуальных компьютерных программ, направленных на то, чтобы понять человеческий интеллект. При этом используемые методы необязательно должны быть биологически подобными.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Обработка естественных языков и понимание речи | | Компьютерное зрение | Самообучение машин | | Автоматическое программирование |
| Нейронные сети | Экспертные системы | Интеллектуальный наставник | Нечеткая логика | | Генетические алгоритмы |
| **Основные приложения ИИ** | | | | | |
|  | | | | | |
| **Дисциплины являющиеся основой ИИ** | | | | | |
| **Лингвистика** | | **Инженерия** | | **Менеджмент** | |
| Психолингвистика, социальная лингвистика, вычислительная лингвистика | | Робототехника, распознавание образов, обработка изображений | | Теория принятия решений, теория организации | |
| **Информационные технологии** | | **Математика** | | **Философия** | |
| Адаптивные системы, ООП | | Исследование операций, статистика, теория управления | | Философия, логика | |

1.7.1 Семиотика и семиотические законы. Семиотика – это наука, изучающая законы построения и функционирования знаковых систем. Семиотика является одним из основания методологии, потому что человеческое общество делает необходимым выработку многочисленных систем знаков. С помощью которых люди могли бы передавать друг другу разнообразную информацию для организации своей деятельности. Чтобы содержание сообщения было понятно получателю, получателю необходим способ с помощью которого получатель смог бы раскрыть смысл данного сообщения. Это возможно только в том случае если сообщение выражается в знаках несущих доверенное их значение и если передающий и принимающий сообщения одинаково понимают связь между знаками и их значениями. Семиотика имеет 3 раздела: 1) Синтаксис, который определяет отношения между знаками и определяет правила построения выражений. 2) Семантика, которая определяет отношения между знаками и объектами и определяет значение знаков. 3) Прагматика – отношение между знаками и пользователями языка.   
1.7.2 Термины и понятия как основа для представления знаний. Виды знаков: 1) Индексы, которые определяют следствие на что-то. 2) Образы, которые формирую подобие на что-то. 3) Сигналы, которые определяют ситуационную связь. 4) Символы (языковые), которые определяют язык. Семантические категории – термины. Они делятся на логические, которые выражают наиболее общие отношения между предметами и ситуациями. Логические в свою очередь делятся на кванторы, пропозиционные связки (типа, или, если-то, иначе, ни-ни) и внутренние связки (не, не равно, не больше). Кроме логических есть ещё нелогические или дескриптивные, которые содержат конкретные значения. Например масса = 20кг и т д. Все выражения разбиваются на различные категории в зависимости от типов их значений и выражаемых символов. Следующий термин Выражения. Они делятся на две группы: 1) Категорематические – предложения и термины. 2) Синтактегорематические – предложения не имеющие определенных типов значений (технические символы и к ним приравненные). Для формирования знаний важное место занимают языки. Языки бывают двух типов: 1) Естественные – формируются стихийно, имеют очень гибкую структуру, выразительно богаты и универсальны. 2) Искусственные – создаются целенаправленно, имеют очень жесткую структуру выразительно ограниченны, узко специализированы. Языковые средства служат для формирования текстов. Тесты бывают повествование, рассуждения и описания.   
1.7.3. Законы семиотики делятся на 3 группы: a) Объективные законы устройства знаковых систем. б) Законы, зависящие от позиции наблюдателя (прагматика). с) Законы смысла, т. е. семантика. Рассмотрим пункт “б” более подробно. В диапазоне систем, которыми пользуется группа людей, любая знаковая система она существует в той или иной мере осознано, среди этой группы людей. В этом случае, когда информационная система не осознается человеком, он может оказаться в одной из трёх позиций: 1) Пользуется языком, но не осознаёт это. 2) Пользуется языком и осознаёт это. 3) Осознает, что перед ним язык, но пользуется им.   
1.7.4. Процедуральные и деклоративные знания. Обычно все знания классифицируют по следующим категориям: 1). Поверхностные – знания о видимых взаимосвязях между отдельными событиями и фактами предметной области. 2). Глубинные – абстракции, аналогии, схемы, которые объясняют процессы предметной области и отображают структуру предметной области. 3). Первичные – были получены с помощью органов чувств. 4). Вторичные – знания, которые выводятся с помощью логического мышления на основании анализа и синтеза. 5). Предметные, которые делятся на процедуральные (процедурные) и деклоративные. Они описывают принципы и порядок предметов в предметной области. Это алгоритмы, методики, инструкции, рецепты, стратегии принятия решений. Процедурные знания составляют процедурную предметную модель обучения. Деклоративные, они представляют собой фактические знания об объектах предметной области. Деклоративные знания определяют содержательную и семантическую часть знаний, эти знания, записанные в память информационных систем таким образом, что они доступны для использования после обращения к памяти с помощью процедур записанных в процедурных знаниях.   
**Раздел 2. Семантика в языках и моделях представления знаний. Спецификация моделей представления знаний, баз знаний и онтологий.**2.1. Семантика – наука о понимании определенных знаков последовательностей символов и других условных обозначений. Виды семантики: 1). Деклоративная – описывает знаки и значения предметной области. 2). Проективная – если есть несколько моделей, то в качестве знаний могут использоваться и предлагаться другие модели из какого-то известного множества. 3). Рефлексивная – это знаки модели с частями моделей.2.2. Критерии полноты, неизбыточности, отсутствия синонимии.   
2.2.1 Свойства информации. Та информация, которая используется в системах управления для принятия решения, должна обладать определенными свойствами: 1). Краткость – минимальный объём, но достаточный для принятия решений. 2). Точность – все измеряемые параметры должны находиться в заданных пределах и с допустимой точностью. 3). Оперативность – вся информация должна фиксироваться в реальном времени. 4). Сопоставимость – эту информацию можно сопоставить с какой-то эталонный информацией. 5). Целесообразность – сбор анализ, принятие решения, выдача управляющих воздействий, всё это должно производиться для достижения цели системы. 6). Рентабельность – затраты на обработку информации должны быть в строго заданных стоимостных ограничениях. 7). Нетендецеозность – независимость от того, кто использует данную информацию. 8). Адресность – информацию должна передаваться ровно туда, где она необходима. 9). Полнота – информация полна, если её достаточно для понимания и принятия решений. 10). Достоверность­­ – информация достоверна если она полностью отражает состояние объекта. 11). Непротиворечивость – она не должна противоречить, полученной с помощью моделирования. 12). Неизбыточность – информации должно быть ровно столько, сколько необходимо для принятия решений. Избыточная информация замедляет процесс выдачи управляющих воздействий. 13). Связанность, вся информация должна быть связана одним критерием – достижение глобальной цели.  
2.2.2. Отсутствие синонимии. Синонимия – это тип семантических отношений языковых единиц, заключающийся в полном или частичном совпадении их значений. Синонимы – это те члены тематической группы, которые: 1). Принадлежат к одной и той же части речи. 2). Настолько близки по значению, что их правильное употребление в речи требует точного знания их семантических оттенков и стилистических свойств. Под синонимией часто подразумевают семантическую эквивалентность. Такая эквивалентность может существовать между словами и словосочетаниями, словосочетаниями и предложениями, предложениями и предложениями. Для установления факта синонимии используется следующие критерии: 1). Совпадение значений. 2). Совпадение синтаксической дистрибуции. 3). Совпадение лексической дистрибуции. 4). Частичная взаимозаменяемость. Самый главный критерий – это первый, все остальные только дополняют его.   
2.2.3. Неизбыточность – это когда каждое значение данных или знаний присутствует в базе в единственном экземпляре. Избыточность может иметь место как на логическом уровне (это когда в структуре данных, повторяются одни и те же типы данных), также и на физическом уровне, это когда данные хранятся в двух и более экземплярах. Принципы интеграции позволяют свести избыточность практически нулевому значению, полной неизбыточности достигнуть практически невозможно. Обычно проверка на избыточность в базах данных и знаний проводится специальными программами.  
2.2.4. Критерий полноты и точности информации. Точной информацией является информация, которая соответствует действительности, подтвержденная независимым источником. Полной информацией является ещё точная информация – это та, которая закончено описывает объект или явление. Критериев оценки точности и полноты различают три: 1). Получив информацию от одного источника, мы обязаны проверить от второго независимого источника. Два источника гарантируют полноту приблизительно 66%. 2). При оценке следует принимать степень откровенности источника, то есть степень надежности аппаратуры съёма и фиксации информации. 3). Надежность источника.  
2.2.5. Критерии неизбыточности: 1). БД неизбыточна, если удаление какого-либо элемента данных ведет к потере информации о предметной области. 2). Избыточность БД проявляется в возникновении дублирования данных или знаний, причем при избыточном дублировании повторяются не просто данные или знания, а информация о предметной области.  
2.3. Представление неполных и неточных знаний.   
2.3.1. Общее положение. При разработки интеллектуальных информационных систем, знания о конкретной предметной области, для которой создаётся система, редко бывают полными и абсолютно достоверными, даже количественные оценки, полученные с помощью точных экспериментов имеют статистические оценки достоверности, надежности и значимости. Большинство знаний, используемых в экспертных системах получены на основании оценок экспертов, мнения которых могут расходиться и являются субъективными. При обработке знаний с применением механизмов формальной логики возникают противоречия между нечеткими знаниями и четкими методами логического вывода. Разрешить это противоречие можно только с помощью преодоление нечеткости знаний или специальных методов нечетких выводов.  
2.3.2. Виды нечетких знаний: 1). Недетерминированность выводов – характерная черта большинства интеллектуальных информационных систем. Она означает, что заранее путь решения задачи в пространстве её состояние найти невозможно, поэтому в большинстве случаев методом проб и ошибок выбирается некоторая цепочка логических заключений, которые согласуются с имеющимися знаниями. А если она не приводит к успеху, организуется перебор с возвратом для поиска другой цепочки. 2). Многозадачность – интерпретации обычно явление, которое используется в задачах распознавания образов, может быть технических систем, так и языковых средств, последнее обычно используется в информационно поисковых системах. 3). Неточность и ненадёжность знаний и выводов – количественные данные, а также знания могут быть неточными, неточность как правило связана с объективными причинами – это в основном несовершенность измерительных приборов, несоблюдение условий проведения замеров. Теория измерений включает в себя способы оценки этих неточностей. 4). Ненадёжность знаний – связана с субъективными причинами: отсутствие нормальных процедур получения точных данных, вероятностная природа поступающих данных. Ненадёжность означает, что для оценки достоверности знаний нельзя применять двухбалльную шкалу. 5). Неполнота знаний и немнотонная логика – абсолютно полных знаний не бывает, поскольку процесс познания бесконечен. В связи с этим база знаний должна изменяться с течением времени. Причем знания, в отличии от данных, могут перечёркивать все знания, которые были до этого. Многие экспертные системы первого поколения были основаны на модели закрытого типа, обусловленной применение аппарата формальной логики. БЗ закрытого типа означает, что все знания системы истины, а все остальные ложны. Добавление данных в БЗ закрытого типа не в коем случае не нарушает полученные до этого выводы.  
2.3.3.Некоторые методы из учёта недетерминированнности рассмотрены в продукционных системах: поиск в глубину, в ширину, принцип метапродукции. В этом случае решение логических или оптимизационных задач пространство поиска представляется в виде графа, именно в виде древовидного графа.  
Метод перебора с возвратами: наиболее известным и часто используемым в том числе и на бытовом уровне является метод перебора с возвратами. Этот метод, метод проб и ошибок, при научном подходе позволяет найти точное решение задачи.

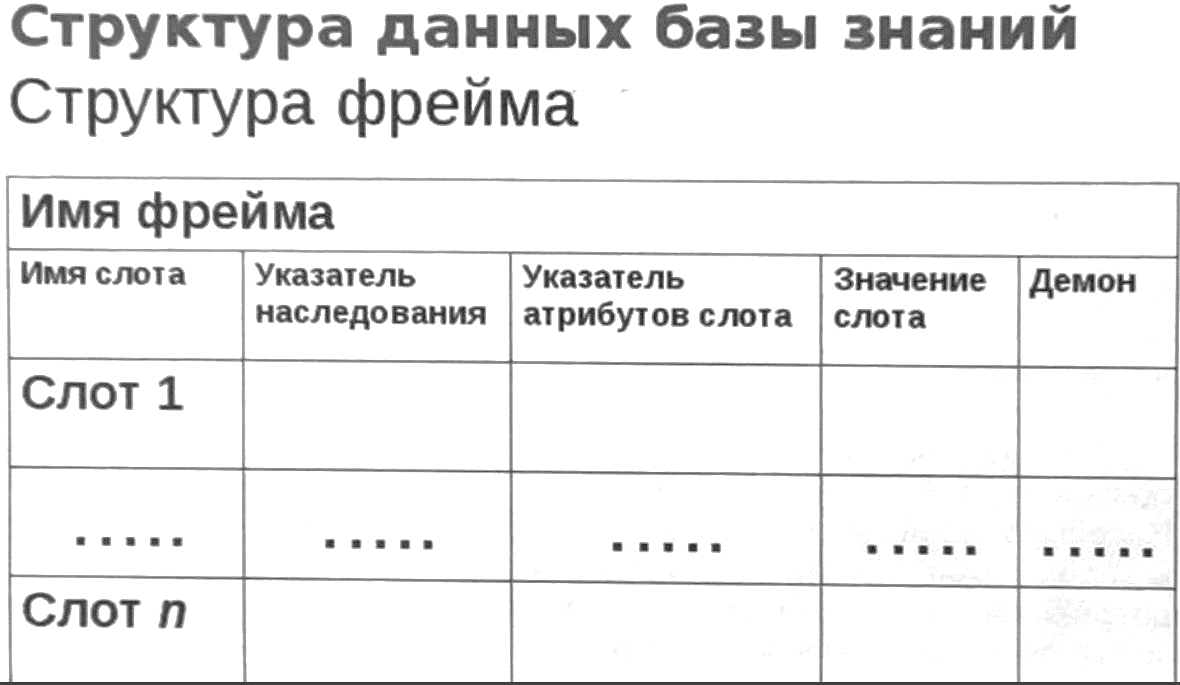


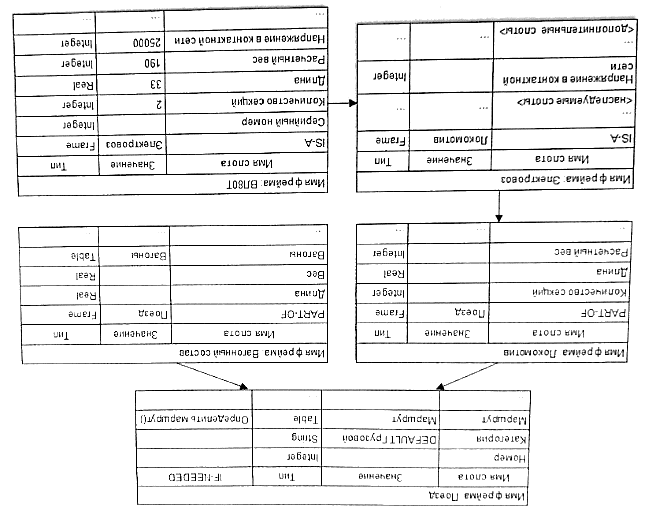
Всё пространство поиска можно представить в виде дерева, узлами которого будут задачи, необязательно верные. Производится последовательно обход этого дерева. В данном случае буквой И обозначены узлы, которые удовлетворяют условиям поиска, Л – узлы, которые не удовлетворяют условиям поиска. Рассмотрим узел 2, если узел 2 истина, то последовательно переходим в 3ему узлы связанному с вершиной 2. Поскольку узел 3 удовлетворяет условию – И, то мы последовательно обходим все входящие или связанные с узлом 3. В данном случае узел 4 и 5 удовлетворяют условиям поиска, т. е. узел 4, 5 могут считаться итоговыми решениями задачи. Узел 6 – Л, происходит возврат к 2му узлу, 2ой связан с 7ым, т. к. он ложен, узлы 8, 9 даже не рассматриваем, переходим к первому и к узлу 10, который Л, поэтому 11, 12, 13 даже не рассматриваем, возврат к узлу 1. В частном случает при обходе дерева, если мы нашли первое решение, то дальнейший обход, может быть прерван. Дерево обходится сверху вниз слева направо и ещё такое метод называется поиск в глубину. Кстати этот метод используется в ЯП Пролог при автоматическом поиске ответа на вопрос. Недостаток этого метода является монотонность выводов, т. е. если в каком-либо узле условие ложно, то в узле нижнего уровня оно не может быть истинным. При большом количестве проверяемых областей этот метод малоэффективный.   
2.3.4.Метод частичного перебора: более точный метод решения оптимизационных задач. Метод ветвей и метод частичного перебора.  
  
При обходе дерева в узлах проверяется соответствие ограничениям и проверяется значение выбранного критерия или целевой функции. В нашем случае у нас стоят критерии минимизации. Оптимальным решением для данного дерева является решение в узле 5, т. к. значение целевой функции меньше чем допустим в 4, из 7го узла поиск не производился, потому что коэф. в узле 3 меньше чем в узле 7, значит узлы 8, 9 дадут заведомо худший вариант, чем узлы связанные с 3 и соотв. с 5. Ограничение этого метода: монотонность критерия или целевой функции (т. е. значение критерия вышестоящего узла, не может быть выше ниже вышестоящего) и монотонность вывода.  
2.3.5.Алгоритм А\* (эвристический метод решения оптимизационных задач). Этот метод не гарантирует нахождение точного или оптимального решения поставленной задачи. Но этот метод даёт достаточно хорошие результаты в большинстве случаев. Эвристические алгоритмы используются в тех случаях, когда точные методы не позволяют найти приемлемое решение задачи за приемлемое время. В нём для всех узлов удовлетворяющих ограничениям и в которые можно попасть из узла, вычисляются значения критерия, дальнейший процесс поиска, выполняется только из узла, в котором значение критерия минимально, остальные ветки поиска не рассматриваются.   


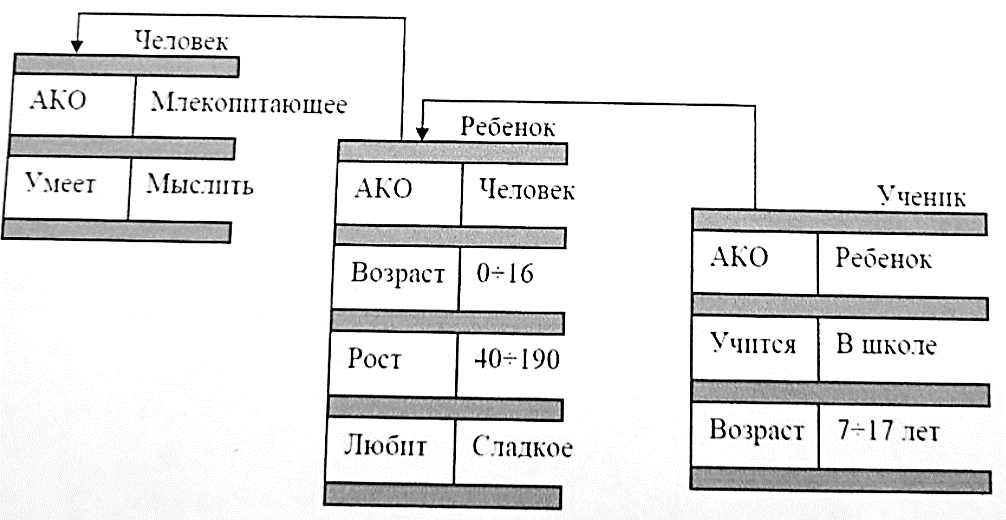
Недостаток – возможный пропуск оптимального решения. Должен был быть выбран узел, у которого коэф. наибольший. Выбран узел 9, а вот узел 10, который должен был быть оптимальным, пропущен, потому что узел 7 имеет коэф. ниже, чем узел 6, и 10 даже не рассматривается. Если сравнить временные характеристики, то самый быстрый метод A\*. Эвристические методы очень широко применяются в игровых задачах, особенно там, где есть игры с соперником.   
2.4.Маштабируемость интеллектуальной системы.   
2.4.1.Структуркая сфема интеллектуальный системы управления.  
Есть какой-то объект управления и на него оказывает влияние внешняя вреда на выбор решений. Среда через внешней источники информации формирует определенные ограничения, обычно результат влияния внешней среды формируется после обработки внешней информации. Поступает входная команда через диалоговое общение, на основании обработанной внешней информации происходит обращение к БЗ и извлекаются определенные знания о внешней среде. Если таких знаний нет, то эти знания заносятся в БЗ. Далее на основании входной информации формируется цель, для того, чтобы правильно сформировать цель, выбирается информация от объектов управления по каналам обратной связи. Это снимается информация с датчиков состояния, обрабатывается внутренняя информация из датчиков, она контролируется и диагностируется и переносится в узел формирования цели. Идёт обращение к БЗ, если необходимой информации их БЗ не получено, то происходит обучение или самообучение. После того как сформированы знания и на основании обратной связи, формируется на основании знаний план действий, которые передаются на соответствующие подсистемы, которые и воздействуют на объект управления.  
2.4.2.Маштабируемость – в электронауке и информатики означает способность системы, сети или процесса справляться с увеличением рабочей нагрузки (увеличивать свою производительность) при добавлении ресурсов обычно аппаратных. Система называется маштабируемой, если она способна увеличивать производительность пропорционально увеличению ресурсов. В системах с плохой маштабируемостью добавление ресурсов лишь незначительно увеличивает производительность, а начиная с обределенного порога увеличение ресурсов не даёт никакого эффекта. Различают: 1). Вертикальное маштабирование –увеличивается каждый компонент системы с целью повышения общей производительности. 2). Горизонтальное маштабирование – разбиение системы на более мелкие структурные компоненты и разнесение из по отдельным физическим машинам или увеличение количества серверов параллельно выполняющих одинаковые функции.  
2.4.3.Методы горизонтального маштабирования. Ускорение программы с помощью параллельных вычислений на нескольких процессорах ограниченно размером последовательной части программы. Есть закон Амдала – в случае когда задача разделяется на несколько частей суммарное время её выполнения на параллельной системе не может быть меньше времени выполнения самого длинного фрагмента. Для горизонтального маштабирования по сути используется 2 метода: 1). Метод распараллеливания программ, который может быть ручным, автоматизированным и полуавтоматизированным. 2). Метод шардирования ­–это горизонтальное партиционирование БД в компьютерном кластере. Партиционирование – это когда независимые строки таблицы БД хранятся раздельно, заранее сгруппированные в секции, которые размещаются на разных физических и логически независимых серверах БД. При этом один физический узел кластера может содержать несколько серверов БД.  
2.5.Базы знаний  
2.5.1.Отличие БД от БЗ. Данное – это представленная в формализованном виде конкретная информация об объектах предметной области, их свойства и взаимосвязь, которые отражают события и ситуации в этой области. Знания – это обобщенная и формализованная информация о свойствах и законах предметной области, для которой реализуется процесс решения задачи. При этом преобразуются данные и сами знания. В этом случае знания играют особую роль по двум причинам: 1). Внешняя часть подобных систем ориентирована на определенный тип знаний, например, на правила и логику. 2). Способ с помощью которого система представляет знания влияет на развитие эффективность, скорость и защиту системы.   
2.5.2.Спецификация БЗ. Спецификация обычно формируется с помощью языка спецификации. Язык спецификации – формальный язык, предназначенный для декларативного описания связей, свойств данных и способов их преобразований без явного упоминания порядка выполняемых действий и использования конкретных значений данных. Спецификация – документ, который точно полностью и в поддающейся проверке форме определяет требования, устройство, поведение или другие особенности системы, компонента, продукта, результата или услуги. Спецификация требований, спецификация структуры, спецификация продукта, спецификация испытаний. Спецификация – это перечисление подробностей на которые необходимо обратить особое внимание. Спецификация – документ устанавливающий требования, стандарт ISO 9000:2015. Детальная инструкция по выполнению работы или по использованию материалов в проекте или инструкция, которая полностью описывает как что-то изготовить. Технологическая инструкция по изготовлению – наше название. Спецификация – это документ описывающий полное описание системы для целей её разработки. ISO 2382-20:1990. Спецификация – документ, который полностью описывает элемент проекта или его интерфейс, а также требования для его приёма и проверки. Аналог у нас – технический паспорт и методика приёма и контроля изделия.   
2.6.Онтология.  
2.6.1.Отличия от философского понятия онтологии. Термин онтология ­– философский термин, но т.к. он используется в информатике, то он принял самостоятельное значение. Есть 2 существенных отличия: 1) Онтология в информатике должна иметь формат, которые формат, который компьютер может легко обработать. 2). Информационное онтологии создаются с конкретными целями – решение конструктивных задач. Эти онтологии оцениваются с точки зрения применимости, а не полноты. Экземпляры (идивиды) это объекты основа нижних уровней компонентов онтологии. Это могут быть абстрактные объекты (числа, слова, текст). Чаще встречаются понятия или классы – коллекции или наборы объектов, они могут включать в себя другие классы, экземпляры или сочетания. Например, понятие люди, оно может быть вложенным в понятие человек, а может быть наоборот. Атрибуты – объекты в онтологии могут иметь атрибуты, каждый атрибут имеет по крайней мере имя и значение и используется для хранения информации, которая специфична для объекта. Например, экземпляр или индивид, есть автомобиль модели А, набор атрибутов, которые его характеризуют: количество дверей, двигатель ([2.2л, 1л]), коробка передач и т.д. Отношение – отношения определяют зависимость между объектами онтологии. Для описания онтологии существуют языки описания онтологий. Онтологии бывают специализированные и общие. Специализированные, или предметноориентированные, это представление какой-либо области знаний или части реального мира. В такой онтологии содержатся специальные для данной области значения терминов. Общие онтологии используются для представления понятий общих для большого числа областей. Такие онтологии содержат базовый набор терминов, которые используются для предметных областей. Если специализированная онтология развивается, то иногда возникает проблема объединений этих отологий. Даже иногда онтологии смежных областей могут быть несовместимы, поэтому чтобы объединить их требуются большие затраты. Несовместимыми они могут быть из-за: использования языковых средств и т.д. Объединение этих онтология происходит вручную или в автоматическом режиме.  
2.6.2.Требования к моделям представления данных. При построении БЗ используются специальные языки для представления знаний основанные на символьном представлении знаний. Эти языки делятся на типы по формальным моделям представления знаний. Основные модели:  
  
Есть 3 требования к представлению данных: 1). Модель должна давать однородное представление знаний. 2). Обеспечить максимальную простоту понимания. 3). Должно быть максимально упрощён механизм управления выводом.

3 раздел Фреймы.

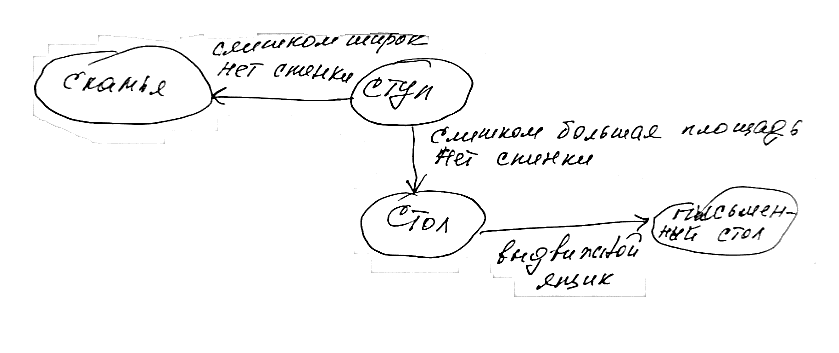
3.1 Представление знаний фреймами. Фрейм – каркас или рамка – термин был придуман и введён в обращение – Марвин Мински 1975г. Фрейм – модель реальной или воображаемой сущности. Фрейм – единица представления знаний, запомненная в прошлом, детали которой при необходимости могут быть изменены согласно текущей ситуации. Фреймы бывают: 1). Фрейм образец – прототип, или протофрейм, – множество сущностей, имеющих общую структуру и поведение. Он выступает шаблоном на основе которого создаются конкретные сущности. 2). Фрейм экземпляр – это конкретная сущность, которая характеризуется индивидуальностью, т. е. индивидуальным состоянием или поведением. Индивидуальность – свойство сущности, с помощью которого её можно отличить от других. Фрейм экземпляр имеет свой уникальный идентификатор. Фрейм экземпляр состоит из слотов, конкретное значение слотов определяет текущее состояние фреймов экземпляров, а набор демонов и присоединённых процедур, а также алгоритмическая реализация определяют поведение фрейма экземпляра. 3). Есть фреймы структуры, которые используются для обозначения объектов и понятий. 4). Фреймы роли. 5). Фреймы сценарии. 6). Фреймы ситуации. Структура данных БЗ. Структура фрейма.

  
В общем случае фрейм задаётся следующими параметрами: 1). Имя фрейма ­– служит для идентификации фрейма и должно быть уникальным. Фрейм представляет собой совокупность слотов. Слот – атрибутивная составляющая или декларативные знания и присоединённые процедуры. Присоединенные процедуры – процедурные знания. 2). Имя слота – идентификатор атрибута сущности, который наделен определенной семантикой. Имя слота должно быть уникальным в пределах фрейма, в качестве имени может выступать произвольный текст или зарезервированное обозначение – системное имя слота. IS-A обозначает это, PART-OF является частью, RELATION относится, AKO это есть. Указатели наследования определяют правила заполнения слота. Они характерны только для фреймовых систем иерархического типа. В конкретных системах указатели могут быть реализованы различными способами: 1). По умолчанию от фрейма образца. 2). Через наследование от родительского фрейма указанного в слоте из IS-A или AKO. 3). По умолчанию или через наследование, но в случае необходимости значение слота может быть изменено, в этом случае используется латинская буква O. 4). Значения слота для всех фреймов экземпляра одного типа должно быть уникальным, ставится буква U. 5). Значение слота для всех фреймов экземпляров одного типа должно находиться в пределе указанном в слоте фрейма образца, ставится R. 1). По формуле указанной в слоте. 2). Через демон. 3). Явно из диалога. 4). Тип Данных (атрибуты) Определяет тип значения слота: фрейм – указатель на фрейм real. Integer – целое число. Boolean – логический тип, text – фрагмент текста, list – список, table – таблица, может быть выражение или присоединённая процедура. 5). Значения слота должно соответствовать указанному типу данных и условию наследования. 6). Демон – IF-NEEDED, IF-ADDED, IF-REMOVED, IF-DELETED. Демон IF-NEEDED запускается, если в момент обращения к слоту его значение не было установлено. IF-ADDED запускается при попытке создания фрейма экземпляра. IF-REMOVED или IF-DELETED запускается при попытке удаления значения слота. 7). Присоединенная процедура – совокупность процедур определяет поведение фреймов, присоединенная процедура демон запускается автоматически или при обращении к этой процедуры из процедуры других фреймов. Процедуры объект управления фреймовой системой.

3.2.Пример фреймового представления данных.  


3.2.1.Организация логического вывода на фреймах  
Наличие иерархической связи позволяет в рамках фреймовой системы организовать логический вывод.   
Пример: Пусть имеется система фреймов.  


Слот АКО – системный слот, значением которого является ссылка на родительский фрейм из которого неявно наследуются некоторые свойства фрейма-образца. Вопрос – любит ли ученик сладкое? Ответ – да. Этот вывод сделан на основании наследования свойства фрейма ребёнок, из которого берётся слот любит сладкое. Может ли ученик мыслить? Да. Вывод сделан на основании наследования слота умеет мыслить человека.

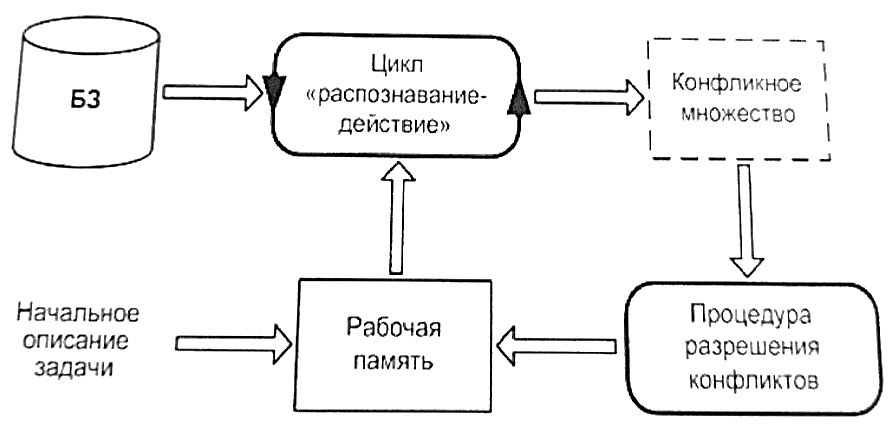
3.3.Свойства фреймов  
1). Иерархическая структура – фрейм обычно представляет собой иерархическую структуру. Особенности такой структуры заключаются в том, что информация о слотах, которые содержит фрейм верхнего уровня, используется всем фреймами нижнего уровня. 2). Межфреймовые сети – осуществление выводов во фреймовой системе возможно благодаря соединению в межфреймовые сети фреймов, которые описывают объекты с небольшими различиями с указателями различия. Например:   
  
3) Значение по умолчанию – фреймы обладают способностью наследовать значения слотов своих родителей. 4). Отношение абстрактное ­­­– конкретное, целое – часть. Отношение абстрактное-конкретное – это на верхних уровнях иерархии располагаются абстрактное объекты, а внизу – конкретные, они регламентируются отношениями типа is A или kind of. Отношение целое – часть касается только структурированных объектов и показывает, что объект нижнего уровня является частью объекта высшего уровня.

3.4.Преимущества и недостатки  
Преимущества: 1). Гибкость. 2). Наглядность. 3). Значение может быть вычислено с помощью процедур или получено эвристическими методами. Недостатки: 1). Высокая сложность системы в целом. 2). Очень трудно внести какие-то изменения в иерархическую систему. 3). Затруднена обработка исключений.

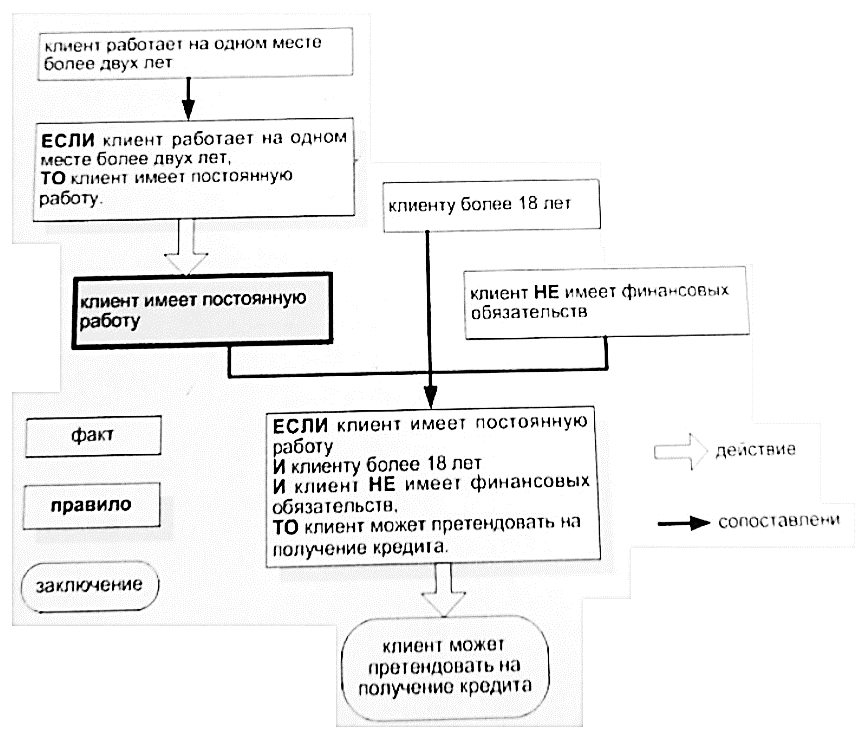
Раздел 4. Продукционные модели.

4.1.Общая структура продукционных правил.  
Продукционные модели были предложены Эмилем Леоном Постом в 1943г. Было проведено исследование решений и принятия действий человеком и сделан вывод, что человек использует продукционные правила. Продукционная модель представления знаний основана на правилах, позволяющих представить знания в виде предложений. Два типа предложений: 1). Если условие, то действие. 2). Если условие A, то действие B, иначе действие C. Но на практике чаще используется первое. В продукционной системе БЗ состоит из БД и Базы Правил. БД содержит факты, описывающие входные данные и состояние системы. База Правил содержит набор продукционных правил в общем виде (первого типа). Условие и действие могут быть связаны логическими союзами И и ИЛИ, а также использоваться НЕ. Например: Если условие1 или условие2, то действие1 и действие2. На практике очень редко можно встретить правила в которых больше двух условий или действий.

4.2.Архитектура продукционной системы.  
БЗ, рабочая память и цикл управления.



Продукционная БЗ состоит из след. ед. 1). Предусловие – условие срабатывания продукции, оно записывается в виде предиката и выполнимость продукции обозначается как логическая истинна. 2). Условие – оно разрешает или запрещает выполнение процедур и устанавливает порядок выполнения. 3). Ядро – состоит из правил. 4). Постусловие – оно содержит рекомендацию по использованию результатов продукции. Моделирование решения задачи основано на процессе сопоставления с образцом в ходе которого текущее состояние решения сравнивается с имеющимися знаниями для определения дальнейших действий. В управляющем цикле распознавание действий осуществляется сравнение образцов из рабочей памяти с условными частями правил из БЗ. Допустимые продукции, т. е. те которые согласованы с текущим сост. рабочей памяти, помещаются в конфликтное множество после этого осуществляется процесс разрешения конфликтов. Разрешение конфликтов производится определенными процедурами цикл распознавания действий происходит до тех пор пока образец рабочей памяти не будет более соответствовать одному из условий рабочей памяти.

4.3.Цепочка вывода  


Эта цепочка говорит как работает цепочка вывода для определенной продукции. Цепочки вывода имеют две разновидности: 1). Монотонный вывод – вывод, при котором факты не удаляются из рабочей памяти. 2). Немонотонный вывод допускает удаление фактов из рабочей памяти. При немонотонном выводе существенную роль играет порядок применения продукционных правил. В процессе вывода решения все правила системы равнозначны между собой и самодостаточны. Самодостаточность – это когда всё необходимое для реализации правила содержится в его условии. И ещё одни правила не могут непосредственно вызывать другие правила.

4.4.Разрешение конфликтов  
Порядок активации правил конфликтного множества определяется выбранной стратегией разрешения конфликтов. Простые стратегии по разрешению конфликтов основываются на том, что выбирается либо первое, либо последнее правило входящее в список. Выбор первого правила соответствует поиску в ширину, а выбор последнего – в глубину. Во многих продукционных системах чаще применяется поиск в глубину, но это были простейшие стратегии. Кроме этого есть и более сложные стратегии разрешения конфликтов. 1). Принцип стопки книг – основан на идее, что наиболее часто используемая продукция является самой полезной. Если возникло конфликтное множество, то выбирается та продукция у которой наибольшая частота использования. Но в этом случае необходимо вести статистику частоты использования, считать как использовалась та или иная продукция. Управление по принципу стопки книг целесообразно применять если продукции независимы друг от друга. 2). Принцип наиболее длинного условия – он заключается в том, что из конфликтного множества выбирается та продукция, у которой стало истинным наиболее длинное условие. Этот принцип опирается на соображение здравого смысла, который говорит о том, что правило которое относится к узкому классу ситуации всегда важнее, чем правило относящееся к широкому классу. Управление этим методом целесообразно применять, когда знания и сами продукции хорошо структурированы по привязкам к типовым ситуациям. 3). Принцип метапродукции – этот принцип основан на идее ввода в систему продукций специальных метапродукций. 4). Принцип классной доски – зачастую принцип классной доски комбинируется совместно с помощью метапродукций. 5). Принцип приоритетного выбора – он связан с ведением статических или динамических приоритетов на продукции. Статический приоритет он обычно присваивается продукции сразу и существует всё время. Динамический присваивается в процессе работы по каким-то определенным правилам. 6). Управление по имени – похожа на стратегию приоритетного выбора, но вместо приоритета задаётся имя продукции. 7). Разбиение на подзадачи, принцип декомпозиции – декомпозиция даёт хороший результат только для хорошо структурированных областей знаний. В этом случае можно разбить задачу на подзадачи, причем разбиение задачи на подзадачи необходимо делать оптимальным способом.

4.5.Направление вывода. 1). Вывод на основе данных. В этом случае процесс решения задачи начинается с исходных фактов, затем применяя допустимые правила, осуществляется переход к новым фактам. И так до тех пор, пока цель не будет достигнута. Этот процесс называется прямой цепочкой вывода. 2). Вывод от цели. Начинается от одной из допустимых целей и рассматриваются пути, которые ведут к достижению цели. Таким образом определяется последовательность правил, позволяющая найти решение. Процесс повторяется для всех заданных задач цели. Такой способ поиска называют обратной цепочкой вывода. 3). Существуют системы для которых вывод основывается на сочетании упомянутых выше выводов. Такой комбинированный метод получил название циклического. Прямая цепочка рассуждений используется в следующих случаях: 1). Когда на основании имеющихся фактов необходимо определить тип объекта или явления, выдать рекомендацию или поставить диагноз. 2). Все или большинство данных заданы в пространстве задачи. 3) Существует большое количество потенциальных целей, но есть только несколько способов представления исходных фактов. 4). Сформировать цель или гипотезы очень трудно в силу избыточности исходных данных или большого количества конкурирующих гипотиз. Обратная цепочка рассуждений применяется в следующих случаях: 1). Для задач соответствующих процессу проверки гипотиз при решении проблем человеком. 2). Цель поиска явно присутствует в постановке задачи или может быть легко сформулирована. 3). Имеется слишком большое число правил, которые на основе исходных фактов продуцируют возрастающее число заключений или целей своевременный отбор цели позволяет отсеять множество тупиковых ветвей. 4). Исходные данные не приводятся в задаче, но подразумевается, что они могут быть известны или легко получены.

4.6.Пример прямого вывода. Пусть в базе правил имеются следующие правила: 1). Прaвило 1 – если двигатель не заводится, то сел аккумулятор. 2) Правило 2 – если указатель бензина на 0, то двигатель не заводится. Предположим что в рабочую память от пользователя поступили факты: 1) фары не горят. 2) бензин на нуле. Рассмотрим алгоритм прямого вывода: 1). Сопоставим факты из рабочей памяти и правила из БП. Правило 1 не срабатывает, а правило 2 срабатывает, т. к. указатель бензина находится на 0, совпадает с условием. 2). Поскольку правило 2 сработало, то в рабочую память заносится заключения этого правила, а именно двигатель не заводится. 3). Второй цикл сопоставления фактов в рабочей памяти с образцами правил. Теперь правило 1 сработает, т. к. двигатель не заводится и фары не горят. 4). В рабочую память заносится действие правила 1, т. е. заключение сел аккумулятор. 5). Конец работы, т. к. БП исчерпана.

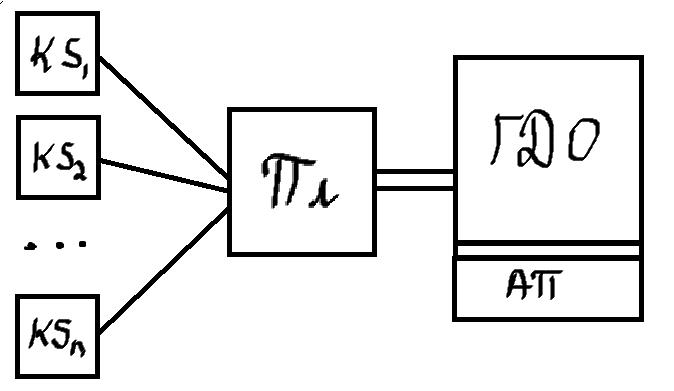
4.7.Пример прямого вывода с конфликтным набором. Пусть в базе правил имеются следующие правила: 1). Прaвило 1 – если двигатель не заводится, то сел аккумулятор. 2) Правило 2 – если указатель бензина на 0, то двигатель не заводится. Теперь допустим, что кроме правила 1 и правила 2, присутствует правило 3 – если показатель бензина на 0, то нет бензина. В рабочей памяти находятся теже правила, что и в предыдущем примере. Предположим что в рабочую память от пользователя поступили факты: 1) фары не горят. 2) бензин на нуле. В результате рабочего цикла может примениться правило 2 и 3, возникает конфликтная ситуации и какое правило применять первым. Если выберем правило 2, то в рабочую память добавится двигатель не заводится. Если выберем правило 3, то мы не найдём вывод. Поэтому нужно будет вернуться к правилу 2. В этом случае результат будет достигнут за 3 шага.

4.8.Пример обратного вывода. Пусть в базе правил имеются следующие правила: 1). Прaвило 1 – если двигатель не заводится, то сел аккумулятор. 2) Правило 2 – если указатель бензина на 0, то двигатель не заводится. Предположим что в рабочую память от пользователя поступили факты: 1) фары не горят. 2) бензин на нуле. Алгоритм обратного вывода содержит следующие шаги. Выдвигается гипотеза. Отыскивается правило, вывод которого соответствует выдвинутой гипотезе, в нашем случае это правило 1. Исследуется возможность применения правила 1, т. е. проверяется может ли оно сработать или нет, но оно сработать не может, потому что в рабочей памяти должны присутствовать факты совпадающие с условием данного правила. Правило 1 не может сработать из-за отсутствия в памяти образца двигатель не заводится, вот этот факт становится новой целью или новой гипотезой. Отыскиваем правило заключение которого соответсвтует новой гипотезе, это правило 2. Исследуется вохможность применения правила 2, оно срабатывает т. к. в памяти присутствует факт совпадающий с его условием. Правило 2 действует и в рабочую память заносится заключение – двигатель не заводится. Условная часть правила 1 теперь подтверждена фактами, следовательно оно сработает и выдвинутая самая первая гипотеза подтверждается. Конец работы.

4.9.Пример обратного вывода с конфликтным набором. Пусть в базе правил имеются следующие правила: 1). Прaвило 1 – если двигатель не заводится, то сел аккумулятор. 2) Правило 2 – если указатель бензина на 0, то двигатель не заводится. Теперь допустим, что кроме правила 1 и правила 2, присутствует правило 3 – если показатель бензина на 0, то нет бензина. Добавим правило 4, если засорился бензонасос, то двигатель не заводится. В рабочей памяти находятся теже правила, что и в предыдущем примере. Предположим что в рабочую память от пользователя поступили факты: 1) фары не горят. 2) бензин на нуле. Алгоритм данного действия: 1) Выдвигается гипотеза сел аккумулятор. 2). Ищем правило заключение которого совпадает с поставленной целью. Это правило 1. 3). Исследуется возможность применения правила 1, оно не может сработать, выдвигается новая гипотеза, двигатель не заводится. 4). Поиск правил заключение которых совпадает с новой подцелью, это правило 2, 4. Если выберем правило 2, то дальнейшее совпадет с примером бесконфликтного набора, а если – 4, то оно не сработает, т.к. в рабочей памяти нет образца засорился бензонасос, после этого опять вернёмся к правилу 2. Следует обратить снимание, что правило 3 не связанное с поставленной целью не затрагивалось в процессе вывода, это свидетельство того, что ненужная продукция просто была пропущена.

4.10.Предусловия и постусловия. Предусловия выражают ограничения выполнения которых необходимо для корректной работы программы. Предусловие применяется ко всем видам и вызовам программ как внутри класса так и у клиента. Корректная программа никогда не вызовет программу, когда невыполнено предусловие. Постусловие выражает свойство или состояние завершающее выполнение программы. Постусловие выражает гарантию того, что программы завершат выполнение и приведут её к состоянию с заранее заданными свойствами.

4.11.Метод доски объявлений.   
Такая методика будет эффективна только тогда, если выполняются следующие требования: 1). Различные модули правил должны иметь одинаковый синтаксис, одинаковые обозначения, одинаковые имена объектов. Это нужно для того чтобы обеспечивать передачу знаний между модулями. 2). Должен существовать протокол определения очерёдности выполнения правил, который вступает в силу тогда, когда сразу несколько модулей готовы изменить содержание доски объявлений. С точки зрения организации процесса вычислений в системе можно выделить следующие структурные компоненты, которые показаны на схеме архитектуры доски объявлений.



ГДО – глобальная доска объявлений. Сюда записываются исходные данные, а также промежуточные и конечные результаты решения задачи. Внизу есть подраздел АП – это активизированные правила. Пл – это планировщик. И КSi – это независимые источники знаний. То есть если источник знаний находит на доске объявлений исходные данные необходимые ему, он формирует запрос через планировщик. Планировщик даёт ему доступ к глобальной доске объявлений. Результаты на доску объявлений передаются через планировщик. Как записан результат доступ у данного источника знаний прекращается. Источники знаний общаются между собой только через доску объявлений и не могут непосредственно передавать друг другу или запускать выполнение процедур. Эта архитектура показала высокую эффективность работы, но модульная организация памяти требует больших ресурсов, поэтому системы построенные на этом принципе работают медленно, тем не менее этот метод считается очень перспективным.

4.12.Таблица принятия решений.  
Таблица состоит из разделов условия, действия и правил. Таблица строится таким образом, что условия должны принимать состояния true или false.



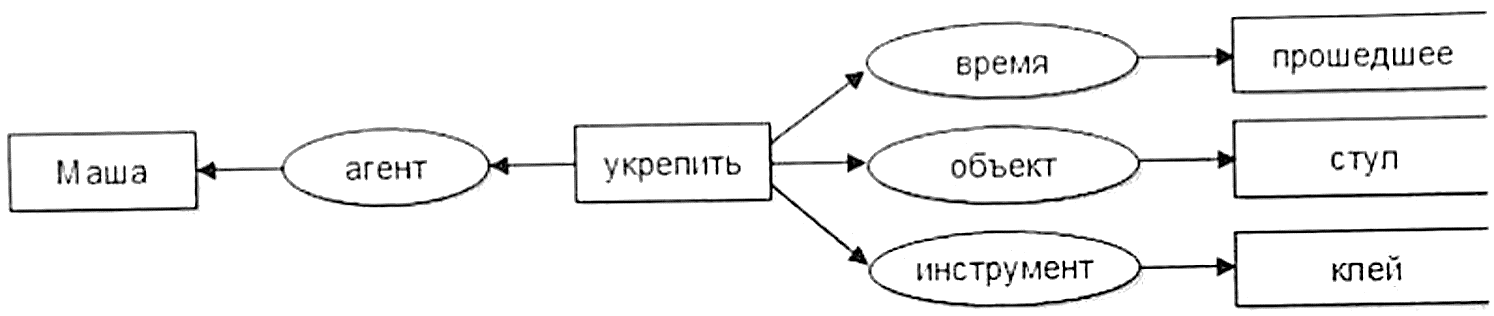
Раздел 5. Семантические сети.   
Семантическая сеть или смысловая сеть – это модель предметной области, представленная в виде графа, где вершины – это понятия, а дуги – отношения между понятиями. В качестве понятий обычно выступают абстрактные или конкретные объекты. В качестве отношений используется следующие отношения согласно смысловой классификации: 1). Таксономические отношения – данный тип отношения определяется отношениями типа АКО или IS, ARE. Таксономические отношения – это класс подкласс экземпляра, или же множество подмножества элемента. 2). Структурные – это отношения типа часть – целое, PART OF. 3). Родовые – это предки – потомки. 4). Производственные – это начальник – подчиненный. 5) Функциональные – определяются глаголами. 6). Количественные. 7). Пространственные. 8). Временные. 9). Атрибутивные – имеющие свойства или конкретное значение. 10). Логические – И, ИЛИ, НЕ. 11). Казуальные (причино-следственные). Отношения также классифицируют по степени участия: 1). Унарные (рекурсивные) 2). Бинарные – связывает два понятия. 3) N-арное ­– отношение связывается более двух понятий. Семантические сети используют в экспертных системах для представления знаний. Есть такая система Проспектор и используют в системах распознавания речи для понимания естественного языка.

5.2.Классификация семантических сетей.  
5.2.1.Классификация. По назначению: включает несколько типов отношений: 1). Классифицирующие. 2). Функциональные – вычислительная модель, позволяющая описывать одни вычислительные единицы через другие. 3) Сценарий. Следующая классификационная единица: 1). Однородные – один тип отношений 2). Неоднородные – количество типов больше одного. Неоднородные сети самые сложные. Следующая классификационная группа – по арности. Последняя по размеру: 1). Для решения конкретных задач. 2). Семантические сети отраслевого маштаба. 3). Глобальная семантическая сеть.

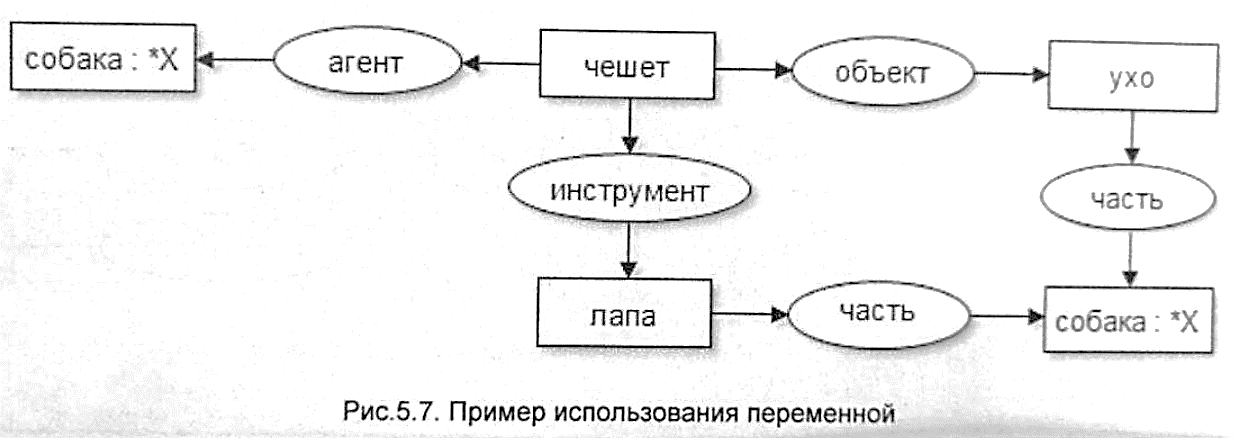
5.2.2.Семантические отношения.   
Основные семантические отношения: 1). Иерархические – отношение типа IS, ARE, AKO, PART OF. 2). Часть чего-то. 3). Вспомогательные – все остальные.

5.3.Достоинства и недостатки семантической сети.  
1). Универсальность – в принципе с помощью семантической сети можно описать сколь угодно сложную ситуацию. 2). Наглядность системы знаний. 3). Близость структуры сети, представляющей систему знаний к семантической структуре фраз на естественном языке. 4). Соответствие современным представлениям об устройстве памяти человека. Недостатки семантических сетей: 1). Сетевая модель не даёт ясного представления о структуре предметной области. Поэтому формирование и модификация такой модели затруднены. 2). Сетевые модели представляют собой пассивные структуры для обработки которых необходим специальный аппарат формального вывода. 3). Проблема поиска решения в семантической сети сводится к задаче поиска фрагмента сети. Этот фрагмент должен соответствовать подсети, которая отражает поставленный запрос. Это в свою очередь обуславливает сложность поиска решения в семантических сетях. 4). Представление, использование и модификация знаний при описании системы реального уровня сложности оказывается очень трудоёмкой процедурой. Особенно при наличии множественных отношений между её понятиями.

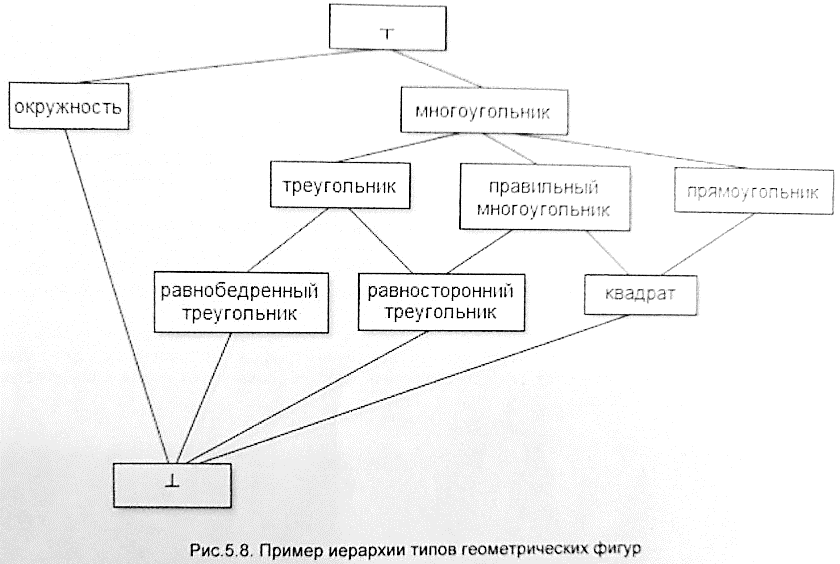
5.6.Концептуальные графы.   
Концептуальный граф – это двудольный ориентированный граф, состоящий из вершин двух типов, понятий и концептуальных отношений. Двудольный граф – это граф множество вершин которого можно разбить на части таким образом, что каждое ребро графа соединяет какую-либо вершин из одной части с какой-то вершиной другой части, но несуществует ребра соединяющего 2 вершины из одой и той же части. Понятия в концептуальном графе отображаются прямоугольниками, а отношения – элипсами. Пример: Маша укрепила стул клеем.



**Маркеры.** Допускается использовать анонимные экземпляры. Анонимные экземпляры обозначаются # и ставится номер, например, собака #1, собака #1 имеет имя Тузик. Разрешается использовать обобщающий маркер. Обобщающий маркер – это \*. Дополнительно к обобщающему маркеру может использоваться переменная, например, собака чешет лапой ухо.

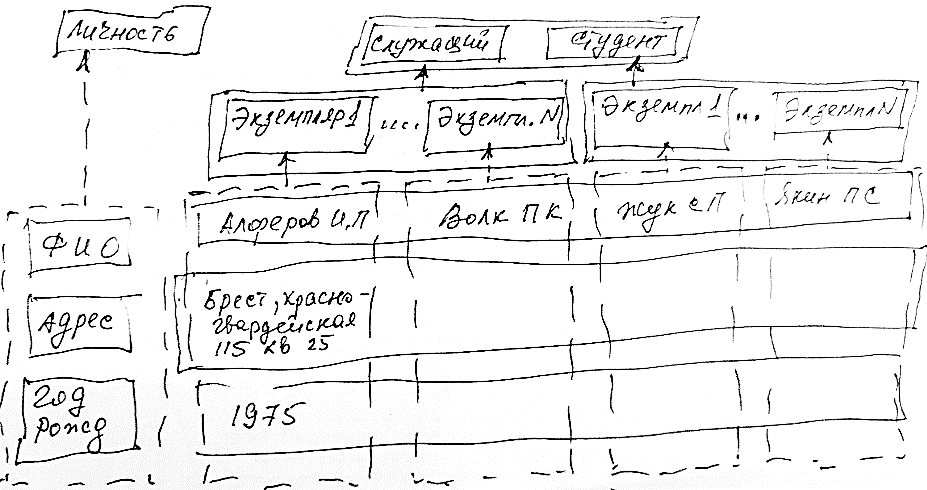


**Иерархия типов.** Отдельнаяиерархия представляется в виде решётки, описывающей таксономические отношения между понятиями, включающие множественное наследование. Для представления иерархии типов в виде решётки в неё включаются два специальных типа: 1). Универсальный тип, который является супер-типом для всех типов и обозначается T. 2). Абсурдный тип, который является подтипом для всех типов и обозначается .

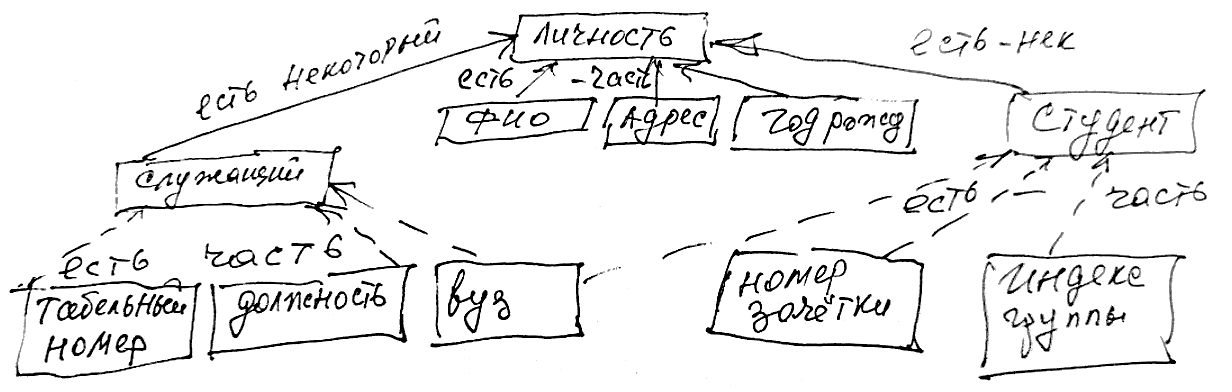
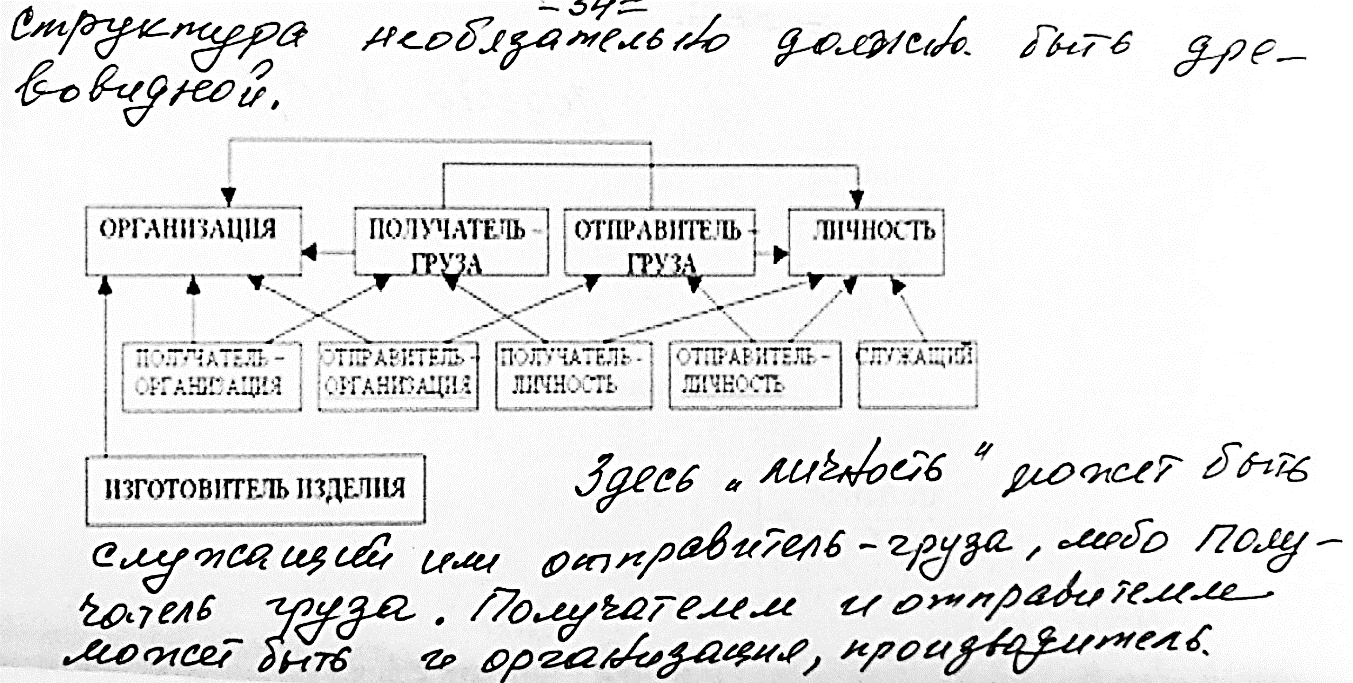


Вилы операций: 1). Копирование, т. е. создание точной копии какого. 2). Объединение двух графов в один, если они имеют общие вершины-понятия. 3). Ограничения – это замена вершины понятия графа другой вершиной понятием из другого графа. 4). Упрощение – исключение дублирующих отношений или понятий.

5.5.Основные концепции Реализуемые в семантических сетях. Их всего 4: 1). Концепция одновременного рассмотрения в модели как знака так и типа. Знак – это конкретное значение или конкретный экземпляр рассматриваемого объекта. Тип – это класс подобных знаков. 2). Концепция иерархии типов – это когда обобщение позволяет соотнести множество знаков или множество типов с одним общим типом. Различают обобщения: знак-тип – это классификация, тип-тип – это и есть обобщение. Процессом обратным процессу классификации является экземпляризация, т. е. порождение экземпляров. А процессом обратным процессу обобщения является специализация. Агрегация позволяет раскрыть структуру объектов. То есть допускается многократное применение агрегации, т. е. любую структуру можно представить как агрегат базовых типов, а сам агрегат, может служить объектом обобщения базовых типов. Агрегация соотносится с понятиями, есть часть или есть некоторый.

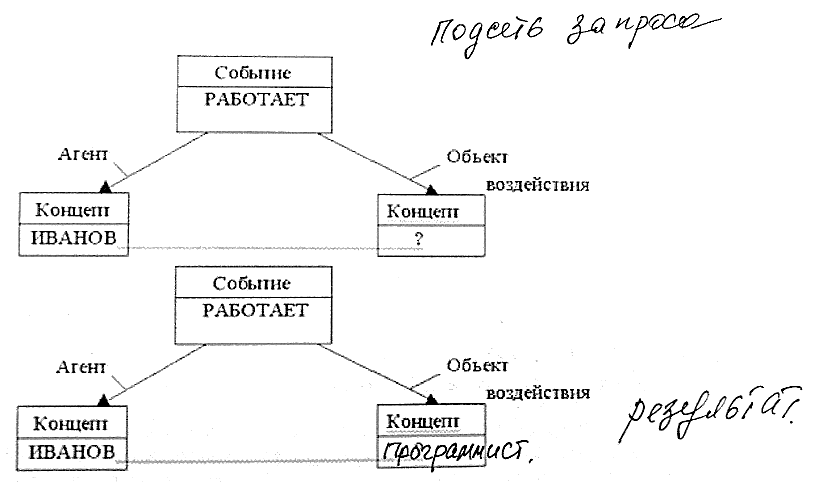
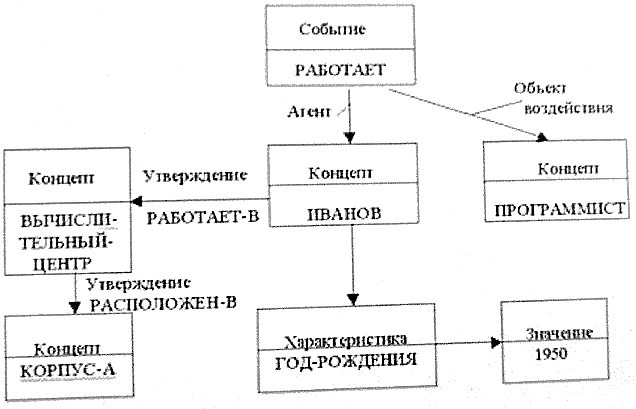


3). Роли. Концепция роли задаётся либо древовидной либо сетевой.  
Древовидная:

  
Сетевая:  


4). Концепция семантического расстояния, эта концепция очень широко используется в словарных системах, где одно слово может быть истолковано посредством других слов, а они посредством третьих и т. д. В этом случае используется мера семантической близости взаимосвязанных понятий. В качестве меры семантической близости взаимосвязанных понятий, если эти понятия представлены в модели вершинами, то той мерой служит число дуг от одной вершины к другой.

5.6.Вывод информации в семантических сетях. Рассмотрим наиболее распространённый подход вывода основанный на сопоставлении частей сетевой структуры. Он заключается в следующем, сначала строится подсеть соответствующая запросу, затем выполняется сопоставление этой подсети с базой знаний. Если управляющая система находит сопоставление это подсети с частью БЗ, то формируется ответ на вопрос.



Раздел 6.Формальные системы и абстрактные машины обработки.

6.1.Формальная система. Формальная система – это совокупность абстрактных объектов несвязанных с внешним миром, в которой представлены правила оперирования множеством символов в строгой синтаксической трактовке без учета семантики. Формальная теория считается определенной если 1). Задано конечное или счётное множество произвольных символов. Конечные последовательности символов называются выражениями теорий. 2). Имеется подмножество выражений называемых формулами. 3). Выделено подмножество формул называемых аксиомами. 4). Имеется конечное множество отношений между формулами называемых правилами вывода. Обычно в теории имеется эффективная процедура, которая позволяет по заданному выражению определить является выражение формулой или нет. Часто множество формул задаётся индуктивным определением и как правило – это множество формул бесконечно. Множество символов и множество формул определяют язык или сигнатуру формальной теории. Чаще всего имеется возможность выяснить является ли данная формула аксиомой. Если да, то теория называется аксиоматической. Множество аксиом тоже может быть бесконечным, если число аксиом конечно, то теория называется конечно аксиоматизируемой. Если множество аксиом бесконечно, то как правило это множество задаётся с помощью конечного числа схем аксиом и правил порождения новых аксиом из схемы аксиомы. Аксиомы делятся на 2 вида: 1). Логические аксиомы – общие для целого класса формальных теорий. 2). Нелогические, или собственные, аксиомы – определяют специфику и содержание конкретной теории. Выводом называется всякая последовательность формул такая, что всякая формула последовательностей есть либо аксиома, либо непосредственное следствие каких-либо предыдущих формул по одному из правил вывода. Формула называется теоремой, если существует вывод, в котором эта формула является последней. Теория, для которой существует эффективный алгоритм, позволяющий узнавать по данной формуле существует ли её вывод называется разрешимой, иначе неразрешимой. Теория, в которой не все формулы является теоремами, называется абсолютно непротиворечивой.

6.2.Разновидности формальной теории.

6.2.1.Дедуктивная теория. Дедуктивная теория считается заданной, если 1). Задан алфавит и правила образования выражений в этом алфавите. 2). Заданы правила образования формул. В этом случае под формулой понимается правильно построенные корректные выражения. 3). Из множества формул некоторым способом выделено подмножество теорем или доказуемых формул или доказуемых формул. Дедуктивные теории имеют разновидности в зависимости от способа построения множества теорем. 1). Задание аксиом и правил вывода. В этом случае в множестве формул выделяется подмножество аксиом и задаётся конечное число правил вывода. Это такие правила, с помощью которых из аксиом и ранее выведенных теорем можно образовывать новые теоремы. 2). В ней задаются только аксиомы, а правило вывода считаются общеизвестными. При таком задании теорем считается, что задана полуформальная аксиоматическая теория. 3). Когда заданы только правила вывода. Множество аксиом пусто, заданы только правила вывода. Заданная таким образом теория – это частный случай формальной теории, но она имеет собственное название – теория естественного вывода.

6.2.2.Свойства дедуктивных теорий. 1). Непротиворечивость. Теория, в которой все формулы являются теоремами называется противоречивой, в противном случае теория непротиворечивая, непротиворечивая теория содержит и аксиомы, и теоремы. Выяснение противоречивости теории – это одна из важнейших задач формальной логики, после того как выяснено, что теория противоречива, то она не имеет ни практического, ни теоретического применения. 2). Полнота. Теория полная, если в ней для любого предложения или замкнутой формулы F можно вывести либо само F, либо его отрицание !F. В противном случае теория содержит недоказуемые утверждения, которое нельзя ни доказать, ни опровергнуть. 3). Независимость аксиом. Отдельная аксиома – независима, если её нельзя вывести из других аксиом. Зависимая аксиома является избыточной и если её удалить, то это никак не скажется на теории. Вся система аксиом теории называется независимой, если каждая аксиома в ней независима. 4). Разрешимость. Теория – разрешимая, если существует алгоритм, позволяющий для любой формулы за конечное число шагов определить является ли она теоремой или нет.

6.3.Типы формальных систем делятся на 4 группы: 1). Логика предикатов – формальная система предметом которой является предложение с учётом их внутренних составом и структурой. 2). Логика высказываний – модель формальной системы, предметом которой является повествовательные предложения взятые целиком без учёта их внутренней структуры. 3) Нечеткая логика – формальная система предметом которой является предложение при нечётком задании характерных признаков отдельных элементов или отношений между ними. 4). Логика отношений – формальная система предметом которой является отношение в виде множества однородных предложений существенно расширяющее логику предикатов. Эту логику ещё называют реляционной.

6.4.Логика первого порядка

6.4.1.Логика первого порядка иногда называется исчислением предикатов. Исчисление предикатов – формальное исчисление допускающее высказывание относительно переменных, фиксированных функций и предикатов. Логика первого порядка расширяет логику высказываний и является частным случаем логики высших порядков. Язык логики первого порядка строится из множества функциональных символов F и множеством предикатных символов Р. С каждым функциональным и предикатным символом связана определенная арность (кол-во аргументов). В языке допускаются следующие символы алфавита: константы, переменные (x, y, …), логические операции (не, конъюнкция, дизъюнкция, импликация (если, то; логически следует), эквивалентность), Кванторы: (квантор всеобщности, квантор существования), служебные символы (cскобки и запятые)

6.4.2.Более сложные конструкции логики первого типа. Основные элементарные символы языка – это атомы. Атомарная функция или формула имеет вид P(t1, t2, …, tn), где P – это предикатный символ арности n, t – термин. Пример: (x + 1) \* (x + 1) >= 0 это атомарная формула, которая действительна для любого x. Формула состоит из двухарного предиката >=. Аргументы этой формулы являются термы и константа 0. (x + 1) \* (x + 1) –это термы, при этом терма состоит из константы 1, переменной x и двух символов бинарных функций +, \*. Терм – конструкция образованная из атомов, скобок и операторов и имеет вид f(t1, t2, …, tn), где f – функциональный символ арности n, t – терм. Формула – это либо атом, либо одна из следующих конструкций: , где F – формулы, x – переменная. Если x не связанна с формулой, её называют свободной в формуле F. Формулу без свободных переменных называют замкнутой формулой, или предложением.

6.5.Аксиоматика и доказательство формул. Система логических аксиом состоит из аксиом исчисления высказываний, их много, но они дополняются двумя аксиомами. . . Аксиоматика определяются аксиомами гильберта, которые тоже дополняются аксиомами: если А и из А следует В, то и В выводимая формула. – правило обобщения, если А выводимо, то и выводима формула для всех аксиом A с переменной x.

6.7.Синтаксис языка логики предикатов.

6.7.1.Три вида атомов. Различают три вида атомов: 1). Константы. 2). Функциональные символы и предикаты. 3). Константы применяются для обозначения объектов реального мира. Константы состоят из одной или нескольких букв алфавита, цифр, других символов за исключением скобок. Например, константа 25, “альбатрос”, “птица”. Функциональные символы обозначают операторы или функции, которые при воздействии на объект возвращают некоторое значение. Например к числу функциональных символов относится “+”, “-“, “цвет”, “возраст”. С помощью предикатных символов задаются отношения между объектами. Например, ”=”, ”>”, ”<”, ”являться”, ”быть\_братом”. Высказывания образуются с помощью этих типов атомов, скобок и операторов. В результате чего получается конструкции которая называется термом. Используя три типа атомов можно получить высказывания трёх типов: 1). Атомарные. 2). Логические. 3). Высказывания с кванторами. Атомарные высказывания состоят их предикатного символа и одного или нескольких термов. (является, кот, абесинец), (быть\_братом, Вася, Петя). Функции тоже могут быть выражены с помощью предикатных символов, например, (=(+2 4) 6), (=(SUM 2 4) 6). Логические высказывания – предложения о которых можно говорить, что они являются истинными или ложными. Логические предложения могут состоять из атомарных выражений, соединенные связками Например, (является, птица, зверь)), (ЕСЛИ ((летает) (откладывает яйца)) (птица)). Высказывания с кванторами могут включать одну или несколько переменных, например, (

6.7.2.Правильно построенная формула ППФ. Формулы, которые построены с помощью логических связок и кванторов, называют ППФ. В вычисление предикатов первого порядка предусмотрено достаточно много правил нормального вывода, которые позволяют строить формулы. Вывод может использоваться для построения новых формул из числа имеющихся в результате применения правил вывода. Пример: ( из этих формул можно вывести, что (. Для доказания истинности выведенных формул, нужно располагать базой знаний и правилами вывода. Всякий раз когда в БЗ обнаруживается высказывание, совпадающее с посылкой некоторого правила вывода, то его заключение добавляется к имеющимся утверждениям. Этот метод вывода называется резолюцией. С помощью правил задающих синтаксис языка нельзя установить истинность того или иного высказывания. Высказывание может быть построено правильно, но оказаться совершенно бессмысленным. Ещё одним важным понятием является процедура унификации, которая позволяет сравнить предложения путём их сопоставления. Говорят, что 2 предложения могут быть унифицированы, если для их переменных существует такая подстановка в результате которых они становятся иденьтичными. Например, пусть у нас есть 2 истинных выражения. (БЫТЬ\_МАТЕРЬЮ, Джейн, Мери), (БЫТЬ\_МАТЕРЬЮ, Мери, Сью). (

Типы языков программирования высокого уровня. Если в качестве признака классификации взять синтаксис образования языковых конструкций, то языки можно разделить на следующие 4 типа: 1). Процедурный. 2). Функциональный. 3). Логический. 4). ООП ориентированный. Процедурный представляет возможность программисту определять каждый шаг в процессе решения задачи. Особенностями таких языков является то, то решение задачи разбивается на шаги и решается шаг за шагом. Программист определяет конструкции для последующей реализации алгоритма. Процедурные языки характеризуются следующими особенностями: 1). Необходимость явного управления памятью. 2). Малая пригодность для символьных вычислений. 3). Отсутствие строгой математической основы. 4). Высокая эффективность реализации на традиционных ЭВБ. Примеры: С, Basic, Assembler, Pascal. Функциональные: одним из первых был язык Lisp. Основная конструкция в таких языках – это выражение. К выражениям относят скалярные константы, структурированные объекты, тела функции, функции. Функциональный язык включает в себя следующие элементы: 1). Классы констант, которыми могут манипулировать функции. 2). Набор базовых функций, которые программист может использовать без предварительного объявления и описания. 3). Правила построения новых функций из базовых. 4). Правила формирования выражений на основе вызова функции. Функциональные языки – языки очень высокого уровня. Логический, или реляционный, язык. К таким относится Prolog – наиболее распространенный и развитый среди логических. Логические языки характеризуются: 1). Высоким уровнем. 2). Высокой ориентацией на логические вычисления. 3). Возможность инверсного вычисления. Переменные в процедурах не делятся на входные и выходные. 4). Логическая неполнота. Логические и функциональные языки – декларативные, в них отсутствует понятие оператор или команда. ООП языки создают окружение в виде множества независимых объектов. То есть каждый объект ведёт себя подобно отдельному компьютеру. Можно использовать как чёрный ящик, не вникая во внутреннее устройство.

6.10.Представление целей и задач в машинах обработки знаний.

6.10.1.Машина обработки знаний включает в себя информационно-поисковую машину и интеллектуальный решатель задач и набор служебных операций обработки знаний. Машина обработки знаний каждой конкретной системы зависит от назначения этой системы. В связи с этим возникает необходимость использования различных машин обработки знаний в зависимости назначения системы. При проектировании машин обработки знаний в интеллектуальных системах возникает ряд трудностей связанных с 1). переносимостью разработанного программного обеспечения на различные платформы. 2). Обеспечение последующей доработки. 3). Универсализацией методов решения задач. В связи с этим возникает необходимость создания универсальной технологии проектирования машин обработки знаний. Эти машины должны обладать следующими свойствами: 1). Универсальность – обработка различных классов задач. 2). Модульность и расширяемость – возможности расширения. 3). Кроссплатформенность – независимость от ОС и аппаратной архитектуры. 4). Параллельность – параллельное использование различных способов решения задач, а также параллельного решения сразу нескольких задач. 5). Обоснованность – решатель задач должен уметь построить алгоритм решения задачи в понятном для пользователя виде.

6.10.2.**Технология проектирования машин обработки знаний** предполагает использование **многоагентной архитектуры** (какая-то законченная структура, которая выполняет какую-либо задачу) при этом все агенты обработки знаний **взаимодействуют между собой** не непосредственно, а через **общую графо-динамическую память**. Это делает машину обработки знаний более гибкой и расширяемой. Общая совокупность агентов, осуществляющих обработку знаний в ИС, обычно различают на основе различных критериев и в основу этих критериев положено 2 способа классификации: **1). На основании внутренней структуры: 1.1) Атомарные агенты** – агент, который не содержит в своём составе другого агента и **Составные агенты** – агент, в составе которого можно выделить ряд более простых агентов. Пример составного агента: агент, который реализует некоторую стратегию решения задачи и имеет в своём составе несколько программных агентов, выполняющих конкретные функции. **2).** **По назначению (6 видов):** **2.1). Поисковые агенты** – осуществляют поиск готового ответа на поставленный вопрос. **2.2). Агенты, реализующие стратегию решения.** К данному классу относятся агенты, которые реализуют принципы решения задач. Обычно эти принципы заложены в определённую стратегию. **2.3). Агенты логического вывода** – агенты, для генерации новых знаний. **2.4). Агенты интерпретации программ** – они предназначены для интерпретации программ, записанных как на внешних, так и на внутренних языках системы. **2.5) Агенты-мусорщики** – предназначены для удаления из памяти информационного мусора, то есть то, что было сгенерировано другим объектом, но утратило свою актуальность. Они ориентируются либо на типовые конструкции, либо нетиповую, но каким-либо образом помеченную. **2.6). Агенты интерпретации знаний и устранения противоречий в БЗ.** Данный класс агентов предназначен для обеспечения корректной интеграции в систему новых знаний, а также для верификации. Агенты данного класса могут активироваться не в конкретный момент решения задачи, а активизироваться при необходимости.

Раздел 7.Специфицированные и специальные представления знаний и онтологии.

7.1.Понятие онтологии. Онтология имеет свою структуру. В общем виде онтология представляет собой набор из 5 категорий: 1). **Классы** – концепты, понятия и категории. 2). Атрибуты – описывают свойта, 3). Отношения. 4). Аксиомы. 5). Экземпляры. Классы или концепты рассматривается как концептуализация всех представлений сущностей или явлений. Они могут содержать в себе другие классы, экземпляры или их отношения. Например, класс мужчина и женщина подкласс класса человек, который подкласс класса млекопитающее. Атрибут – свойство или концепт, каждый атрибут наделён смыслом или значением. Атрибут содержит информацию присущей данному существу или явлению. Значение атрибута может быть сложным типом. Например, автомобиль, двигатель может быть представлен списком, который заключается в скобки. Отношение – тип взаимодействия между концептами. Используются родовые отношения, отношения включения, категоризации, is-a, класс-подкласс, общее-частное. Отношения как и концепты могут основывать иерархию. Отношения быть\_отцом и быть\_материю для для множества людей может быть включено в отношение быть\_родителями, а это уже может содержаться в отношении быть\_предком. Функции – это тогда, когда n-ый элемент отношения однозначно определяется (n-1)-ым предшествующим элементом. Аксиомы – определяют условия соответствий понятий или отношений и используются для записи высказываний, которые всегда истинны и очевидны. Экземпляры (индивиды) – индивидуальные сущности, отдельные представители класса. Все элементы онтологии организованы в иерархию. Нижняя ступень – экземпляры.

7.2.Классификация онтологий и ресурсов онтологического типа. Онтологии и ресурсы онтологического типа могут существенно различаться по ряду параметром, что позволяет классифицировать их по ряду свойств. Свойства: 1). **Степень формальности онтологии.** Все онтологии в которых таксономию соблюдают свойство транзитивности называются **формальными**, иначе **неформальными**. 2). **Содержимое и цель создания онтологии**. В отеественной науки выделяют 4 типа онтологии по цели создания: A). **Онтологии представления**. Цель их разработки – описание знаний и создание языка представления знаний, а также создание языка для спецификации онтологий более низкого уровня. Например, WOL. Б). Онтологии верхнего уровня. Цель создания – создание единой правильной онтологии, которая включала бы знаний общие для всех областей и которые можно использовать многократно. В). **Онтологии предметной области.** Цель описание знаний предметной области. Подобные онтологии могу повторно применяться в рамках предметной области для которой они были созданы, кроме этого они делятся на 2 группы: **Базовые онтологии (содержат основные концепты)** и **Узконаправленные онтологии**. Г). **Прикладные онтологии**. Цель: описание концептуальной задачи, конкретно. Они могут быть использованы только для этой задачи. 3). Объём и тип структуры концептуализации и её предмет. A). **Общие онтологии**. Б). **Онтологии задач**. В). **Предметные онтологии**. Г). **Онтологии представления** (необходимы для описания онтологии более низкого уровня). Д). **Онтологии задач в предметной области**. Е). **Онтологии методов**. Определяют понятия и связи описывающие механизмы работы для выполнения конкретной задачи. Ж). **Прикладные**. Для конкретного приложения. 4). **Степень проработки концептуализации**. 5). **Количество и качество понятий**. 6). **Наличие экземпляров**.

7.3.Проектирование онтологий. Сегодня не существует единственного правильного метода проектирования онтологий, но поговорим о некоторых рекомендуемых правилах. 1). Не существует единого правильного способа моделирования предметной области, т. е. всегда существует альтернатива. Лучшее решение всегда зависит от ожидаемых расширений системы. 2). Разработка онтологии – это обязательно итеративный процесс. 3). Понятия в онтологии должны быть близки к понятиям и отношениям в предметной области, после того, как онтология будет построена, она может быть отлажена или предъявлена группе экспертов. Этот процесс будет продолжаться в течении всего жизненного цикла онтологии. Р**екомендуемый алгоритм создания онтологии**. 1). **Определение области и маштаба онтологии**. Для этого нам необходимо ответить на следующие вопросы: Какую область будет охватывать онтология? Для чего мы будем использовать данную онтологию? Кто будет использовать и поддерживать данную онтологию? На какие типы вопросов должна давать ответы информация в онтологии. 2). **Рассмотреть варианты повторного использования уже существующих онтологий**. 3). **Перечислить важные термины в онтологии**. 4). **Определение классов и иерархии классов**. То есть необходимо организовать классы и таксономию, а также создать иерархию: класс 🡪 подкласс 🡪 вид 🡪 группа. 5). **Определение свойств классов**. Описать слоты. В слотах описывают свойства классов. 6). **Определение фацетов слотов**. Слоты могут иметь различные фацеты, которые описывают тип знания, разрешенные значения, число значений и другое свойства, которые может принимать слот. Например, вино\_а, тогда фацеты которые описывают этот слот будут, имя, тип, кардинальность, объём и т.д. Мощность слота определяет сколько значений может иметь слот, минимальное значение должно равняться единице. Если максимальная мощность равна 0, то для определенного подкласса не может иметь значений. Тип слота: строка, число, boolean, нумерованные (список конкретных значений), экземпляр. 7). **Создание экземпляров**. 7.1). **Что должно быть в экземпляре?** Например, вино\_а1 должно иметь следующие слоты: крепкость: легкая, цвет: красный, вкус: мягкий, уровень танина: низкий, виноград: аркацетерий, производитель: масандер. 7.2). **Определяется правильность иерархии классов**. 7.3). **Проверяется транзитивность иерархических отношений**. Если Б подкласс A, а C подкласс Б, то С подкласс А. 7.4). **Анализ узлов братьев в иерархии классов**. То есть узлы братья в иерархии, которые являются подклассами одного и того же класса. Например, класс А в него входят подкласс Б и С, Б и С – узлы браться. Если класс имеет только один подкласс, то скорее всего при моделировании допущена ошибка либо онтология неполная. Если у класса более 10 подклассов, то возможно нужно ввести промежуточную категорию. Категория нужна, чтобы сгруппировать эти 10 подклассов на более мелкие группы. 7.5). **Должно быть осуществлено множественное наследование**. 7.6). **Рассмотрение обратных слотов**. Обратные слоты – это когда значение одного слота может зависеть от другого.

Раздел 10.

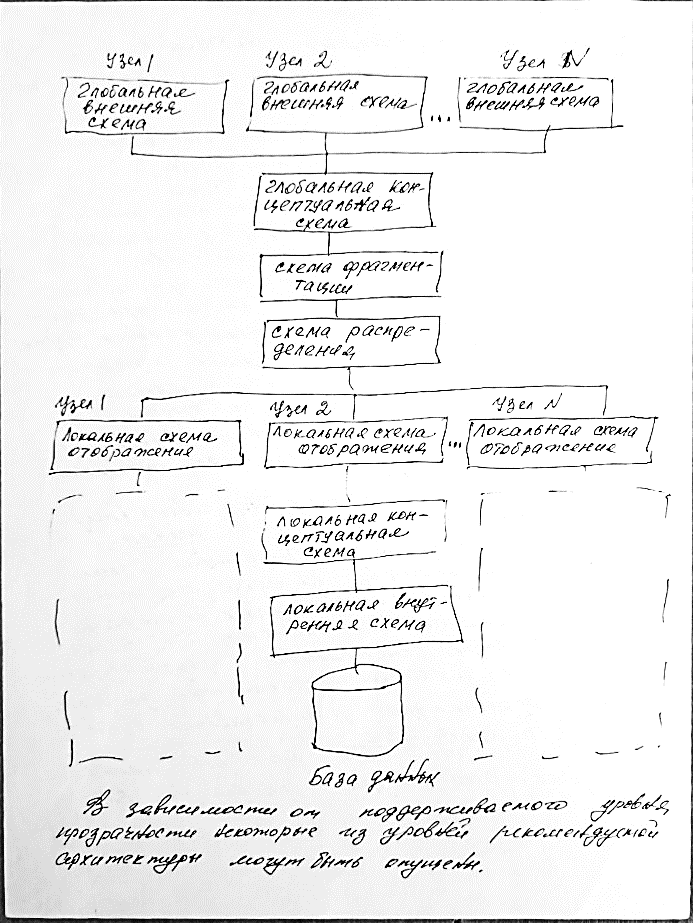
10.1.Основные понятия: 1). Распределенная БД – набор логически связанных между собой совокупностей разделяемых данных и их описания, которые физически распределены в сети. 2). Распределенная СУБД –это программный комплекс, который предназначен для управления распределенными БД. 3). В распределенной СУБД должно существовать хотя бы одно глобальное приложение. 4). Нет необходимости в том, чтобы на каждом из узлов системы существовала своя собственная локальная база. 5). СУБД должна выполнить распределение данных прозрачным и незаметным для конечного пользователя. 6). Цель обеспечения прозрачности состоит в том, чтобы распределенная БД выглядела как централизованная. Кроме распределенной БД есть ещё распределенная обработка. То есть в это случае БД централизована, а доступ к ней осуществляется с разных ПК. Существует ещё параллельная СУДБ – когда функционирует с помощью нескольких процессоров и жестких дисков, что позволяет распараллеливать некоторые операции.

10.2.Преймущества и недостатки распределенной СУБД. Преимущества: 1). Распределенная СУБД отражает структуру организации. 2). Очень высокая степень разделяемости и локальной автономности. То есть данные могут быть перемешены именно в тот узел, которым чаще всего пользуются пользователи. 3). Повышение доступности данных. В централизованных СУБД отказ центрального компьютера прекращает функционирование системы, при отказе одного узла в распределенной системе приводит только к отказу этого узла. 4). Повышение надежности. Если организованно копирование данных, то данные могут размещаться на нескольких узлах. 5). Увеличение производительности. 6). Модульность системы. Расширение системы намного проще, чем при централизованной системе. 7). Экономические выгоды. Если БД распределена, значит на каждом узле может быть свой набор данных с которым она работает. Недостатки: 1). Повышение сложности. 2). Увеличивается стоимость. 3). Возникают проблемы защиты данных. 4). Усложняется контроль над целостностью данных. 5). Резко усложняются процедуры разработки БД. Потому что кроме обычных процедур, нужно принимать решения о фрагментации данных, распределение фрагментов по различным узлам, определение правил копирования.

10.3.Гомогенные и гетерогенные распределенные СУБД. Гомогенные – когда все узлы системы используют один и тот же тип СУБД. Гомогенные системы проще всего проектировать сопровождать и эксплуатировать, такие системы позволяют поэтапно наращивать размеры системы, добавляя новые узлы к уже существующим. Организуя на некоторых узлах параллельную обработку, можно повысить производительность системы. Гетерогенные – когда могут функционировать различные типы СУБД, использующие разные модели данных. Гетерогенные системы возникают при интеграции в существующую систему независимых узлов со своими собственными БД. В гетерогенных системах для организации взаимодействия необходимо организовать трансляцию передаваемых сообщений. Пользователи должно вводить запросы для той СУБД которая используется. В общем случае в гетерогенных системах данные могут быть запрошены с другого узла, которые имеют иной тип используемого оборудования, иной тип СУБД, иной тип оборудования и СУБД. Если используется иной тип оборудования и одна СУБД, то трансляция заключается в замене кода и замене длины кода. Если различны типы СУБД, то трансляция отображается изменение в структуре данных одной модели в структуру данных другой модели. Кроме этого необходимо транслировать язык манипулирования данными из одного в другой. Если отличается оборудование и СУБД, то необходимо выполнить оба вида трансляции описанных выше. В состав отдельных гетерогенных систем включают шлюзы для перевода из одного языка в другой.

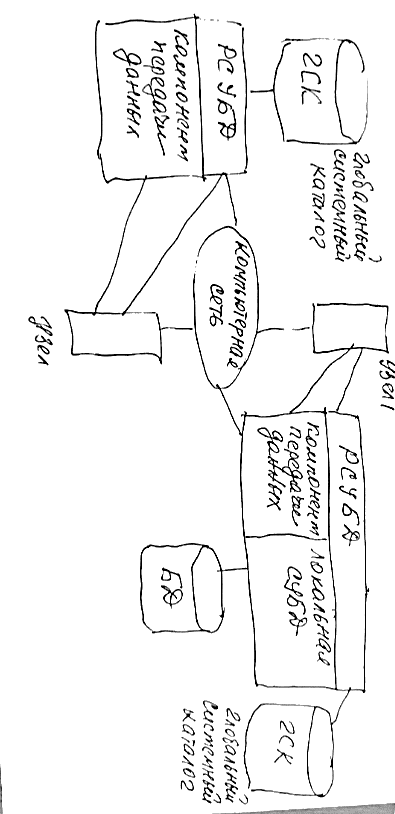
10.4.Архитектура распределенных СУБД

10.4.1.Общая архитектура.



В зависимости от поддерживаемого уровня прозрачности некоторые из уровней распределенной структуры могут быть опущены. Глобальная концептуальная схема представляет логическое описание всех БД, которой она представляет как будто БД не является распределенной. Этот уровень содержит определение сущностей, связей, требований защиты и ограничений поддержки целостности информации. Логическую независимости данных обеспечивают глобальные внешние схемы. Схема фрагментации содержит описание того, как данные должны распределяться по разделам. Схема распределения является описанием того где расположены имеющиеся данные. Схема распределения учитывает все использованные в системе копирования. Каждая локальная СУБД имеет свой собственный набор схем.

10.4.2.Компонентная архитектура



А). Локальная СУБД – компонент, представляющий стандартную СУБД для управления данными в каждом узле. Локальная СУБД имеет свой системный каталог или словарь, в котором содержится информация о данных, хранимых в этом узле. Б). Компонент передачи данных – ПО позволяющее всем узлам взаимодействовать между собой. Он содержит информацию о линии связи и узлах. В). Глобальный системный каталог – выполняет те же функции, что и системный каталог, но кроме этого он содержит специфическую информацию для распределенной системы: фрагментации и распределения. Г). Компонент распределенной СУБД является управляющим элементом по отношению ко всей системе.

10.7.**Разработка распределенных реляционных БД**. Есть несколько стратегий при разработке реляционных БД. 1). **Централизованное размещение** – создание на одном из узлов единственной БД под управлением СУБД доступ к которой будут иметь все пользователи сети. При такой системе уровень затрат на передачу высок, уровень надежности низок. 2). **Раздельное фрагментированное размещение**. В этом случае БД разбивается на непересекающиеся фрагменты. Каждый фрагмент размещается на одном из узлов. При отсутствии копирования стоимость хранения данных будет минимальная, но при этом будет невысок уровень надежности и доступности к данной системе. Однако надежности выше чем в первой стратегии. При правильном размещении данных уровень производительности системы будет относительно высок, а затраты на передачу – низок. 3). **Размещение с полной репликацией**. Эта стратегия предусматривает размещение полной копии всей БД на каждом из узлов системы. В таких случаях надежность, доступность, уровень производительности будет максимальными. Однако стоимость устройств хранения данных, уровень затрат на передачу информации об обновлениях БД Будут самыми высокими. Для преодоления этих проблем используют технологию снимков. Снимок представляет собой копию БД в определенный момент времени. 4). **Размещение с избирательной репликацией**. Данная стратегия представляет собой комбинацию методов фрагментации, репликации и централизации. В данном случае одни данные разделяются на фрагменты, а остальная часть хранится централизованно. На практике именно данная стратегия чаще всего используется.

10.8. **12 правил Дейта, которым должна соответствовать распределенная СУБД.** С точки зрения обычного пользователя распределенная система должна выглядеть точно также как и нераспределенная – это называется основной принцип. 1). **Локальная автономность** – узлы в распределенной системе должны быть автономными, то есть локальные данные принадлежат локальным владельцам и сопровождаются локально. Все локальные операции остаются сугубо локальными. Все операции на заданном узле контролируются только этим узлом. 2). **Отсутствие зависимости от центрального узла** – в системе не должно быть ни одного узла без которого она не могла бы функционировать. 3). **Непрерывное функционирования** – работа не должна прерываться при добавлении или удалении узла из системы или при динамическом создании или удалении фрагментов на одном или нескольких узлов. 4). **Независимость от местоположения** – правило эквивалентно прозрачности местоположения. Пользователь должен получать доступ к БД с любого узла независимо где этот узел находится. 5). **Независимость от фрагментации** – пользователь должен получать данные независимо от способа их фрагментации. 6). **Независимость от репликации** – пользователя не должно волновать наличие копий. 7). **Обработка распределенных запросов** – система должна поддерживать обработку запросов, которые ссылаются на данные расположенные в нескольких узлах. 8). **Обработка распределенных транзакций** – система должна поддерживать транзакцию как единицу восстановления. Система должна гарантировать, что глобальные и локальные транзакции будут выполняться с сохранением 4 основных свойств транзакции: атомарность, согласованность, изолированность и долговечность. 9). **Независимость от типа оборудования** – распределенная СУБД должна функционировать на оборудовании с различными платформами. 10). **Независимость от ОС**. 11). **Независимость от сетевой архитектуры**. 12). **Независимость от БД**. Последние 4 правила на данный момент являются почти недостижимыми.

Раздел 10. Фрагментация

10.9.1.**Назначение фрагментации**. 1). **Условие использования** – с точки зрения размещения данных целесообразнее организовывать работу приложения с определенными подмножествами отношений. Эти подмножества можно рассматривать как минимальную единицу размещения. 2). **Эффективность** – все данные хранятся в тех местах в которых они чаще всего используются. В узле исключается необходимость хранения данных которые не используются локальными приложениями. 3). **Параллельность** – поскольку фрагменты являются минимальными единицами размещения, то и транзакции могут быть разделены на подзапросы. 4). **Защищенность** – данные не используемые локальными приложениями не хранятся в узлах. Пользователи не обладающими правами доступа не получат доступ. Недостатки фрагментации: 1). **Производительность** глобальных приложения, использующих множество фрагментов, ниже, чем локальная. 2). **Целостность** – поддержка целостности усложняется. Т. к. различные фрагменты могут размещаться в разных узлах.

10.9.2.**Типы фрагментации**. Три типа: 1). **Горизонтальная фрагментация** – когда таблица делится на несколько фрагментов. Каждые фрагменте – это разбиения таблицы, которые имеют абсолютно одинаковую структуру. 2). **Вертикальная фрагментация** – разбиение таблицы га фрагменты по подмножеству атрибутов отношений, то есть по столбцам. 3). **Смешанная фрагментация** – совмещение двух предыдущих, горизонтальная фрагментация подвергается ещё и вертикальной, или наоборот, если была произведена вертикальная, то эти фрагменты далее подвергаются горизонтальной.