

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 29.07.2022 17:51:01

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

РАЗДЕЛ 1

МЕТОДЫ ПОИСКА И ПРИНЯТИЯ РЕШЕНИЙ

Лекция 1.1 – Методы полного перебора в ширину и глубину

Интеллектуальная система отличается от любой другой тем, что в ней в качестве обязательных составляющих должны присутствовать: модель предметной области; модели механизмов мышления, то есть метапроцедур, работающих на системе знаний, представленных моделью предметной области (в частности, процедур логического вывода); естественно+языковой интерфейс, обеспечивающий взаимодействие пользователя с интеллектуальной системой.

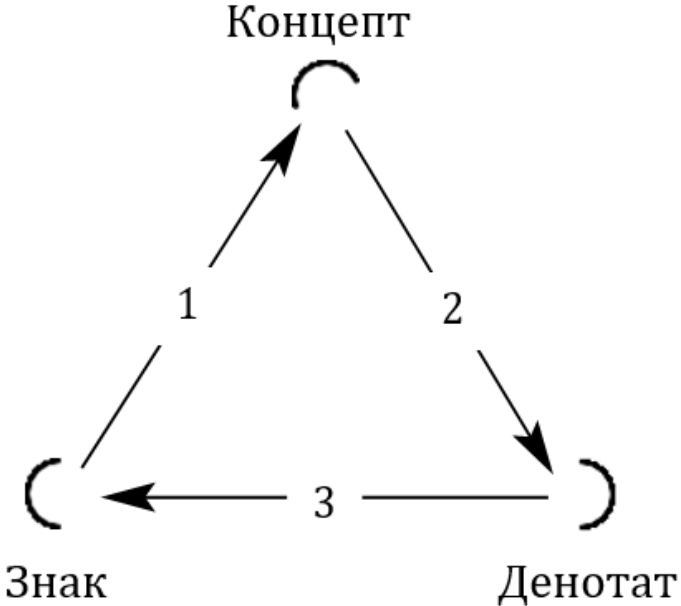
Искусственный интеллект (ИИ) (AI, Artificial Intelligence) – совокупность метапроцедур – представления знаний, рассуждений, поиска релевантной информации в среде имеющихся знаний, логического вывода, пополнения знаний, их корректировка и т.д. – то есть процедур, имитирующих мыслительную деятельность человека.

Система искусственного интеллекта (СИИ) – аппаратный и информационно: программный комплекс, действие которого аналогично действию механизмов мышления человека и неотлично от решений, которые принимались бы человеком - экспертом, то есть профессионалом в данной предметной области.

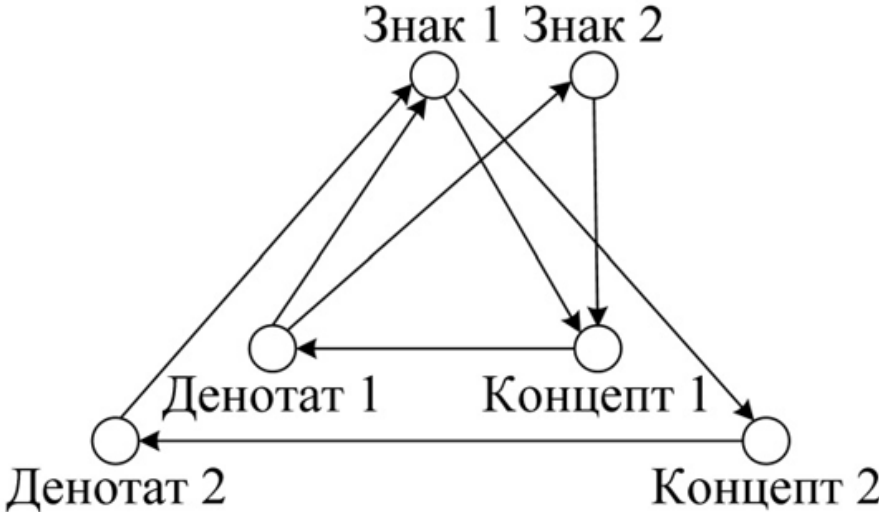
Лекция 1.1 – Методы полного перебора в ширину и глубину

Под **системой знаний** понимается совокупность знаний, образующих целостное описание некоторой проблемы с доступной и достаточной степенью точности.

Знание обладает как минимум пятью обязательными свойствами: **внутренней интерпретируемостью**, **структурированностью** (связностью), **вложимостью**, **погруженностью** в пространство с «семантической метрикой», **активностью**.



Треугольник Фреге



Знаковые ситуации

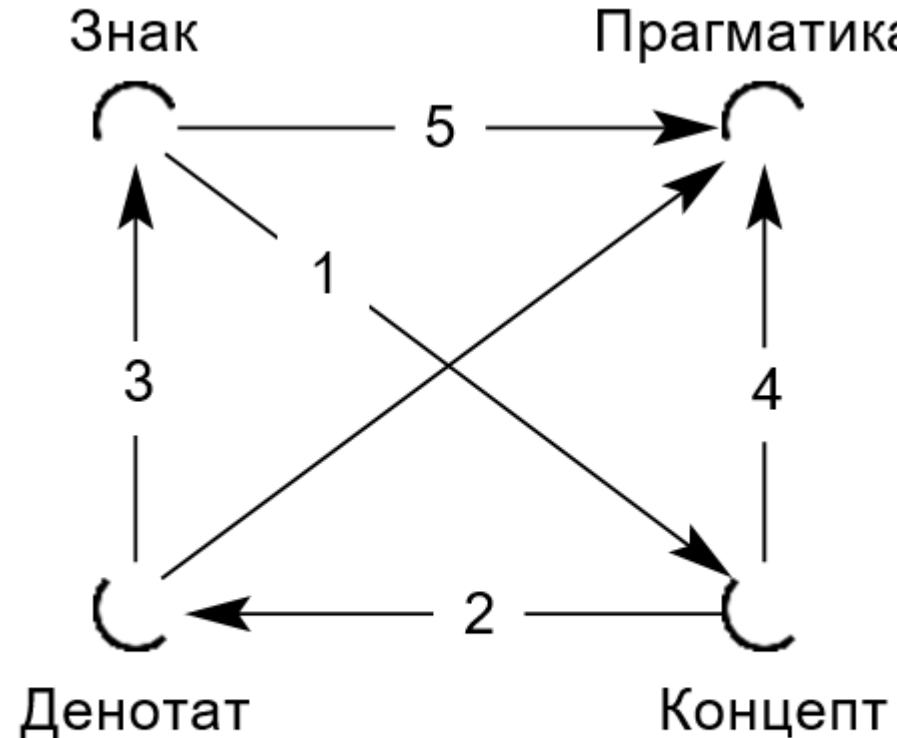
Лекция 1.1 – Методы полного перебора в ширину и глубину

Любая **знаковая система** имеет три аспекта изучения: **семантика**, **синтактика**, **прагматика**

Семантика. Это отношение знака к обозначаемому, то есть его концепт и денотат вместе

Синтактика. Это аспект знаковой системы, связанный с правильностью построения знака из более простых.

Прагматика. Это тот аспект знака, который относится к восприятию этого знака адресатом.



Четырехугольник Поспелова

Лекция 1.1 – Методы полного перебора в ширину и глубину

Задача заключается в том, чтобы перевести предметную область из состояния S_H в некоторое заданное, определяемое как целевое S_U

Сделать это возможно, лишь применяя допустимые в данной предметной области действия из множества $G = (g_1, g_2, \dots, g_k)$

Процедура решения задачи

$$Z = (S_H \xrightarrow{G} S_U)$$

Путь поиска решения можно отобразить следующей цепочкой

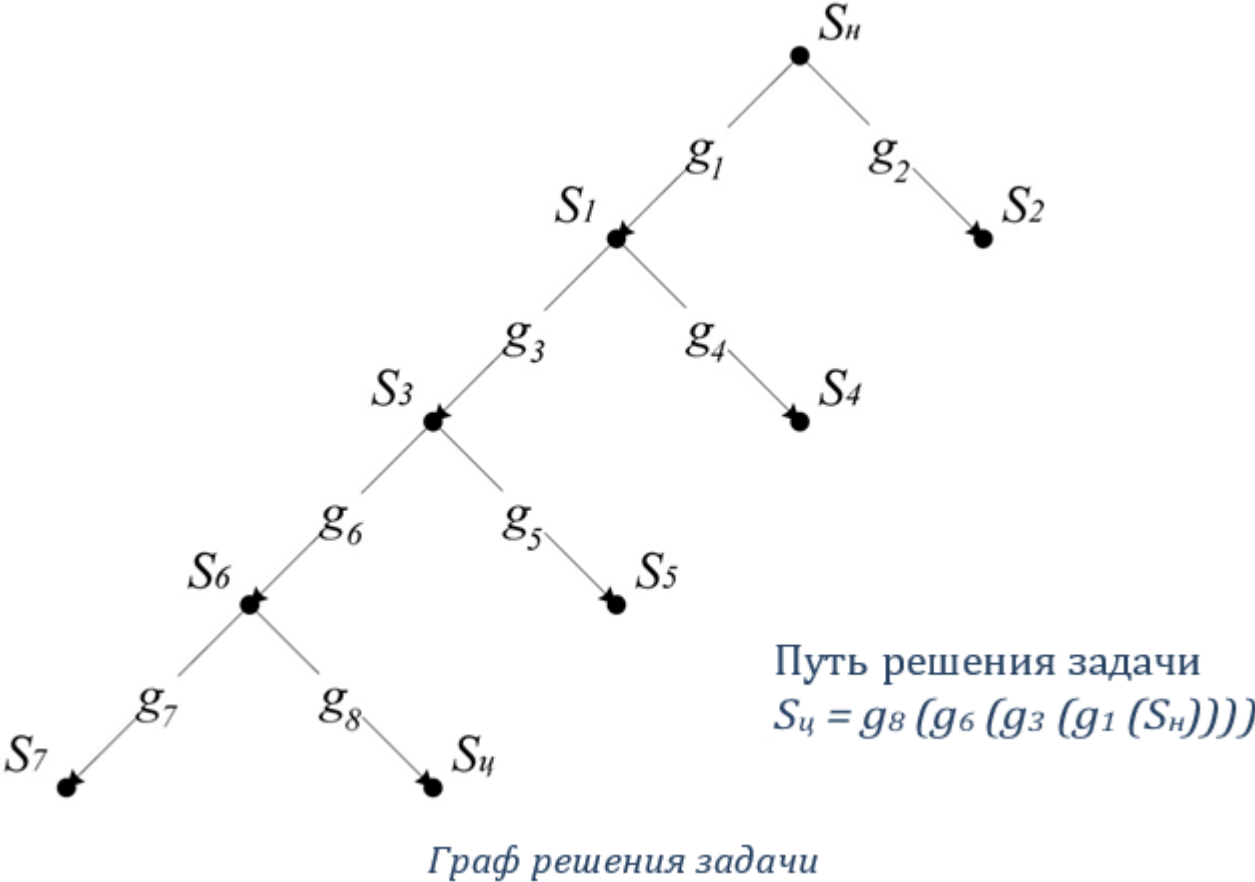
$$S_H \xrightarrow{g_1} S_1 \xrightarrow{g_2} S_2 \xrightarrow{g_3} \dots \xrightarrow{g_{j-1}}$$

$$S_U = g_j (g_{j:1} (\dots (g_3 (g_2 (g_1 (S_H))) \dots))).$$

Лекция 1.1 – Методы полного перебора в ширину и глубину

Множество возможных состояний предметной области назовем пространством состояний. Процесс этот может быть длинным и сложным, поэтому возникает необходимость в некотором едином методе представления множества состояний и поиска решений. Таким методом является метод, удобной графической моделью которого стал граф.

Если отождествить состояние S_n с корнем или начальной вершиной графа-дерева, то, применяя к S_n какой-либо оператор $g_1 \in G$, мы порождаем новое состояние S_1 , образуя тем самым следующую вершину графа



Лекция 1.1 – Методы полного перебора в ширину и глубину

Общая процедура построения дерева в пространстве состояний

- 1) К корню дерева S_n применяются операторы g_i из множества G (их может быть несколько). Полученные при этом вершины образуют первый уровень новых вершин.
- 2) Каждая из вновь полученных вершин проверяется, не является ли она целевой. Если нет, то процесс продолжается по отношению к каждой из них. Образуется второй уровень вершин.
- 3) Если к какой-либо вершине никакой оператор из G не применим, то эта вершина становится терминальной (конечной). Как видим, на каждом шаге проводятся две операции: порождение новой вершины и проверка, не является ли новая вершина целевой, то есть совпадающей с целевым состоянием.
- 4) Когда целевая вершина найдена, в обратном направлении (от цели к началу) просматриваются указатели дуг и выделяется путь решения. Практически этот путь удобнее отображать посредством операторов, связанных с этими дугами.

Лекция 1.1 – Методы полного перебора в ширину и глубину

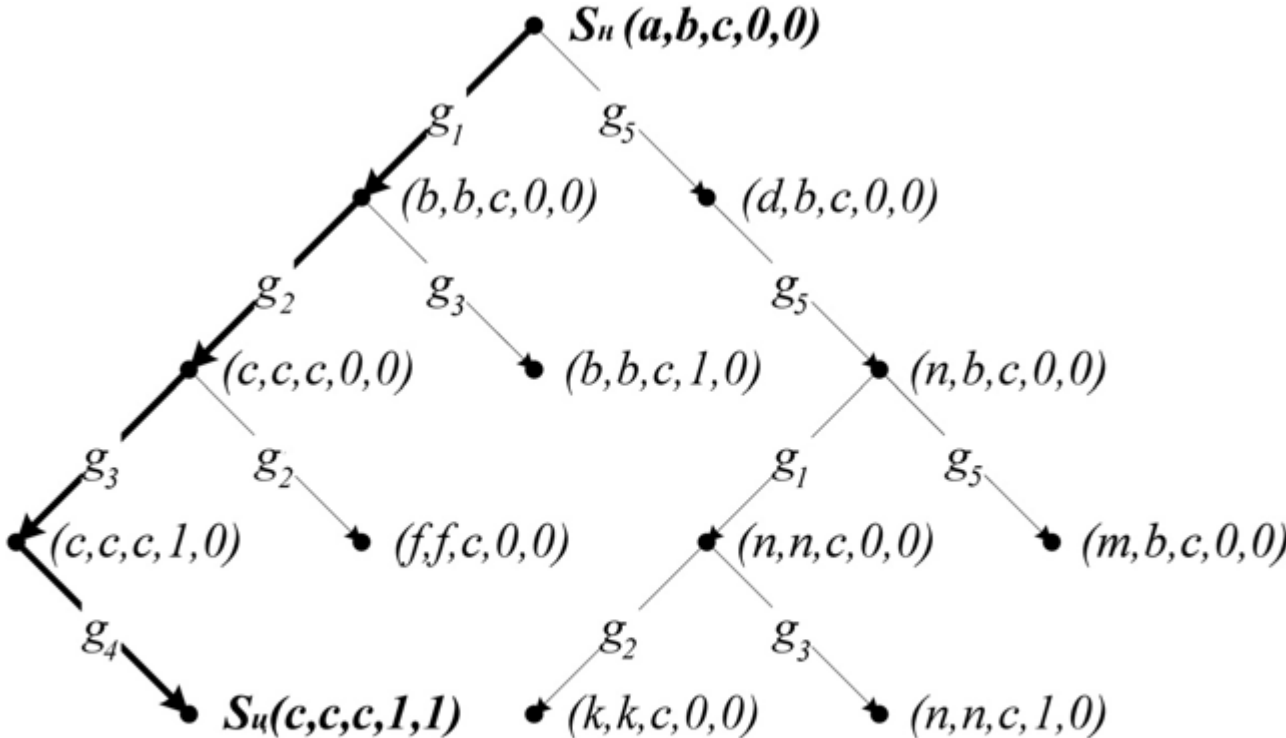
Проблема перебора вершин

Варианты:

- 1) если вершины раскрываются в том же порядке, в котором они порождаются, то такой процесс называется полным перебором в ширину (breadth-first process)
- 2) если на каждом шаге первой раскрывается вершина, которая была построена последней, то такой процесс называется полным перебором в глубину (depth-first process).
- 3) если есть некоторая дополнительная (эвристическая) информация о предметной области, которая позволяет делать суждения о характере графа пространства состояний и расположения цели, то такой метод построения графа называется эвристическим («эвристический» означает «служащий открытию»).

Лекция 1.1 – Методы полного перебора в ширину и глубину

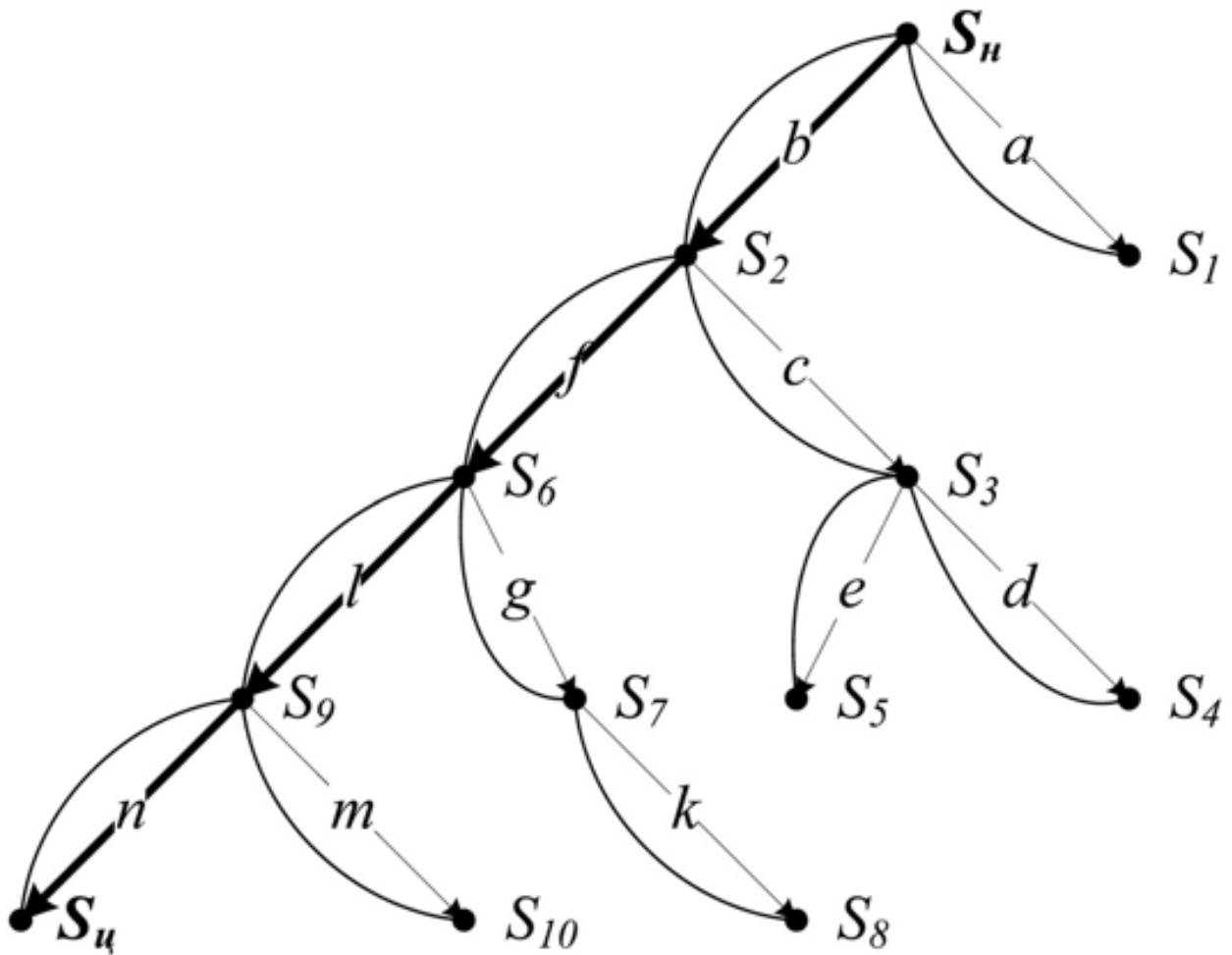
Метод **полного перебора в ширину** гарантируют нахождение целевой вершины как раз потому, что перебор – **полный**



Элемент дерева полного перебора в ширину для примера с обезьяной и бананами

Лекция 1.1 – Методы полного перебора в ширину и глубину

Общая схема перебора в **глубину**



Дерево полного перебора в глубину

Лекция 1.1 – Методы полного перебора в ширину и глубину

Алгоритм перебора в глубину

- 1) Раскрывается начальная вершина, соответствующая начальному состоянию S_H .
- 2) Раскрывается первая вершина, получаемая в результате раскрытия S_H . Ставится указатель.
- 3) Если она раскрывается, то следующей будет раскрываться вновь порожденная вершина. Если вершина не раскрывается, то процесс возвращается в предыдущую вершину.
- 4) По получении целевой вершины процесс раскрытия заканчивается и по указателям строится путь, ведущий к корню. Соответствующие дугам операторы образуют решение задачи.
- 5) Если для заданной глубины раскрытия целевая вершина не находится, то весь процесс повторяется снова, а в качестве новой вершины рассматривается самая левая из полученных на предыдущем этапе.

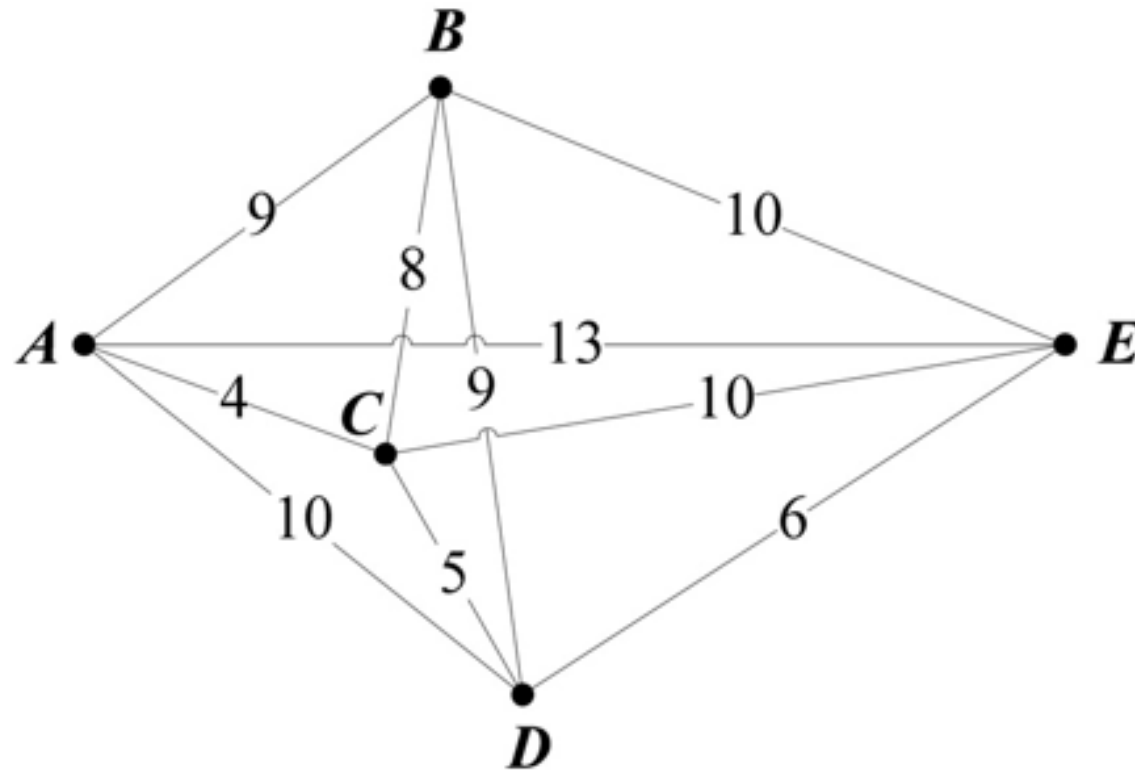
Лекция 1.2 – Эвристические методы поиска в пространстве состояний

Методы полного перебора гарантируют решение задачи, если оно существует, а при наличии нескольких решений, гарантирует оптимальное. Однако на практике эти методы используются для решения лишь небольших по размерности графов состояний задачи. Для реальных случаев чаще всего используется дополнительная информация, основанная на предыдущем опыте или полученная на основании теоретических выводов

Такая информация называется *эвристической*, а организованная в правила – *эвристическими правилами* или *эвристиками*. Эвристическая информация носит сугубо специальный характер и может применяться только в рамках данной задачи, в лучшем случае, в рамках задач данного класса

Лекция 1.2 – Эвристические методы поиска в пространстве состояний

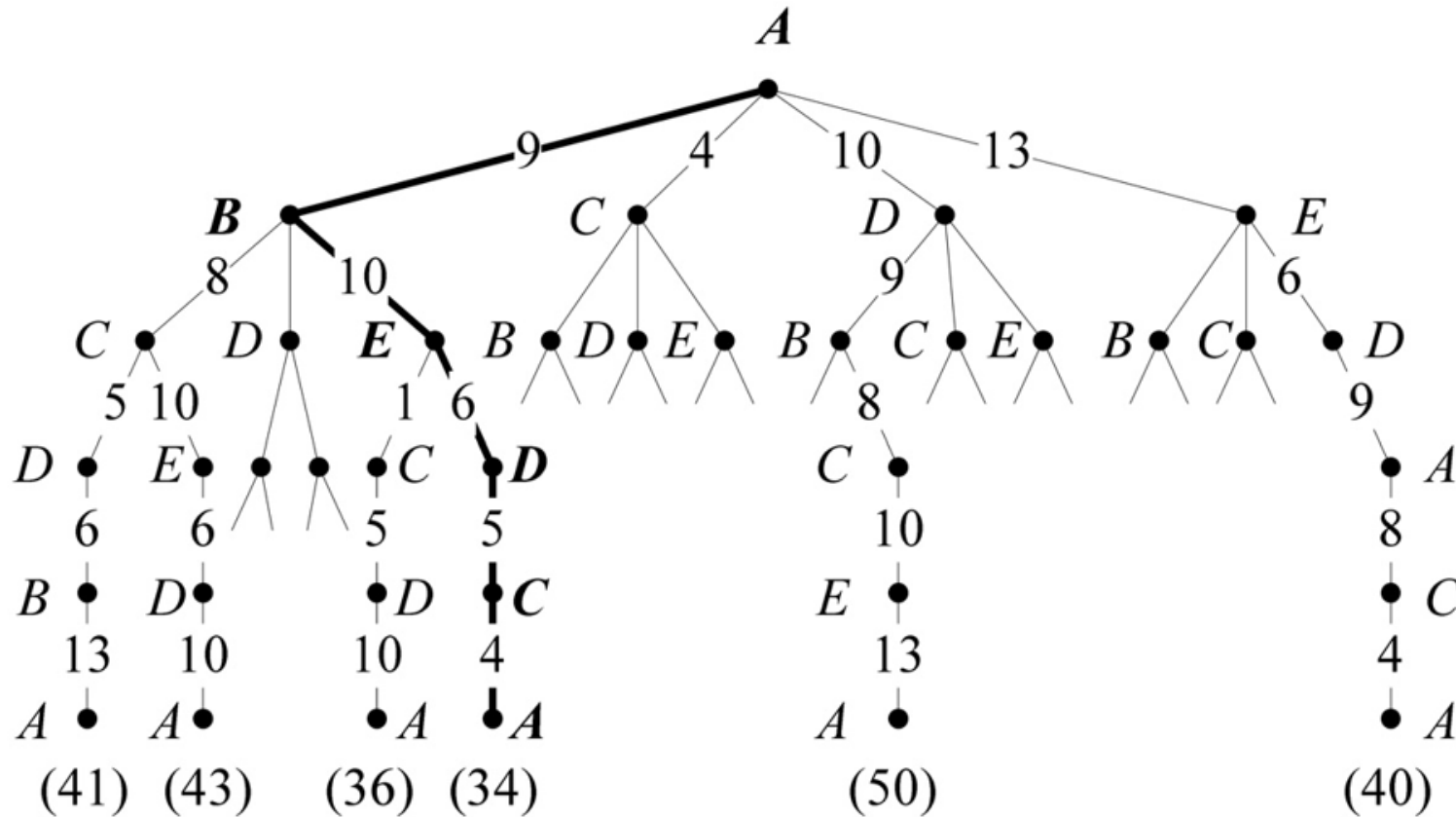
Пример эвристического поиска – известная задача о коммивояжере, $n=5$, длины путей на ребрах графа



Задача о коммивояжере

Лекция 1.2 – Эвристические методы поиска в пространстве состояний

Пусть начальным пунктом отправления будет город A , поэтому $S_n = A$.
Целевое состояние – тоже A . Оператор перехода – единственный – g_1

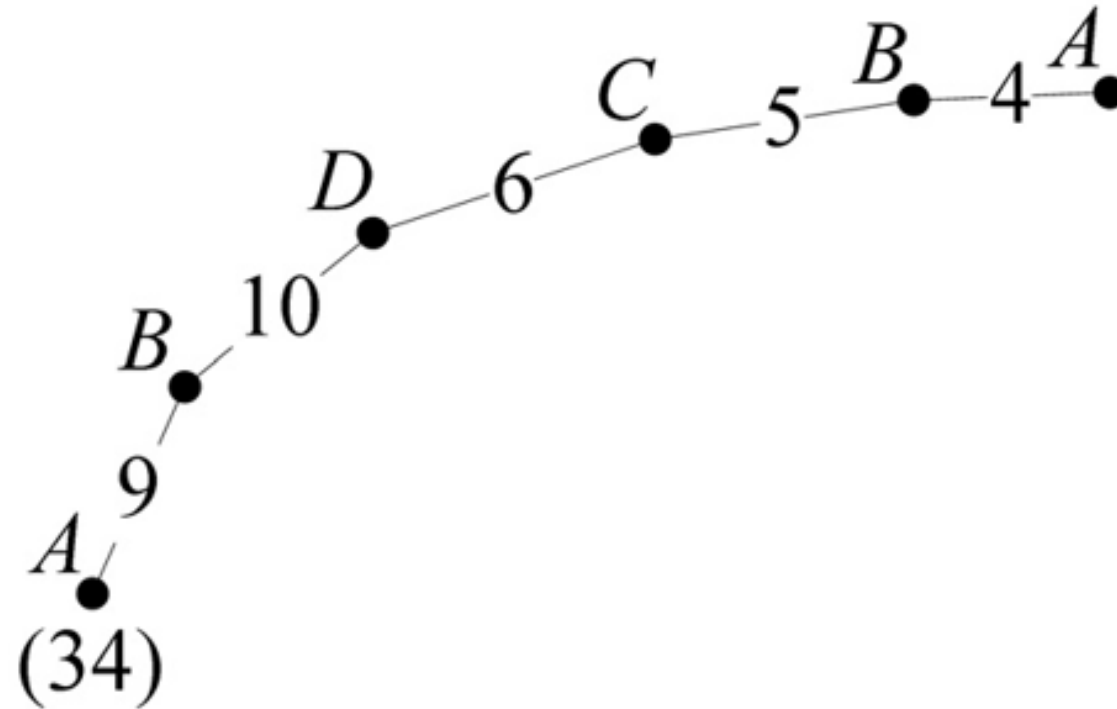


Граф полного перебора в ширину для задачи о коммивояжере

Лекция 1.2 – Эвристические методы поиска в пространстве состояний

Граф полного перебора, включающий все возможные пути коммивояжера, будет содержать 24 варианта

Длина оптимального пути – 34 км ($ABEDCA$)



Граф поиска решения задачи в глубину

Лекция 1.2 – Эвристические методы поиска в пространстве состояний

В общем случае эвристические методы не гарантируют нахождение оптимальных решений, но очень к ним близких. Зато получается большая экономия в затратах на поиск решения, что бывает очень важным фактором.

Существует много методов применения эвристических оценочных функций. Процедура перебора на заданную глубину часто называется также программированием с обратным слежением (back-track programming).

Возможно также использование комбинированных стратегий поиска методами и в ширину, и в глубину: перебор этапами, ограничение числа дочерних вершин, двунаправленный перебор от S_n к S_u и обратно, методы ветвей и границ и динамическое программирование, применяемые в исследовании операций

Лекция 1.3 – Оценочные методы принятия решений

Исследования процессов принятия решений человеком, выполненные методами психологии, нейрофизиологии и другими научными дисциплинами, показали, что психологический механизм регулирования актов поведения тесно связан с построением в структурах мозга информационной модели той среды, в которой находится организм, включая также сведения о целевом условии и возможных способах воздействия на среду, необходимых для решения той или иной задачи. Целевые условия представляют собой некоторое желаемое состояние внешней среды, то есть то состояние, в которое должна перейти среда в результате воздействия субъекта или системы



Лекция 1.3 – Оценочные методы принятия решений

Концептуальная модель представляет собой некоторый набор, или алфавит образов и моделей реальной и прогнозируемой ситуаций и совокупность знаний о возможных управляющих и исполнительных действиях. Содержание образов и моделей реальной ситуации должно соответствовать задачам, стоящим перед решающим конкретную проблему человеком. Основными механизмами регулирования процесса построения решения являются операции: распознавание и классификация ситуаций, формирование гипотезы решения, прогнозирование возможного результата решения и др. Сам процесс принятия решений основан на модели объекта управления или, употребляя другую терминологию, на его ситуационной абстракции



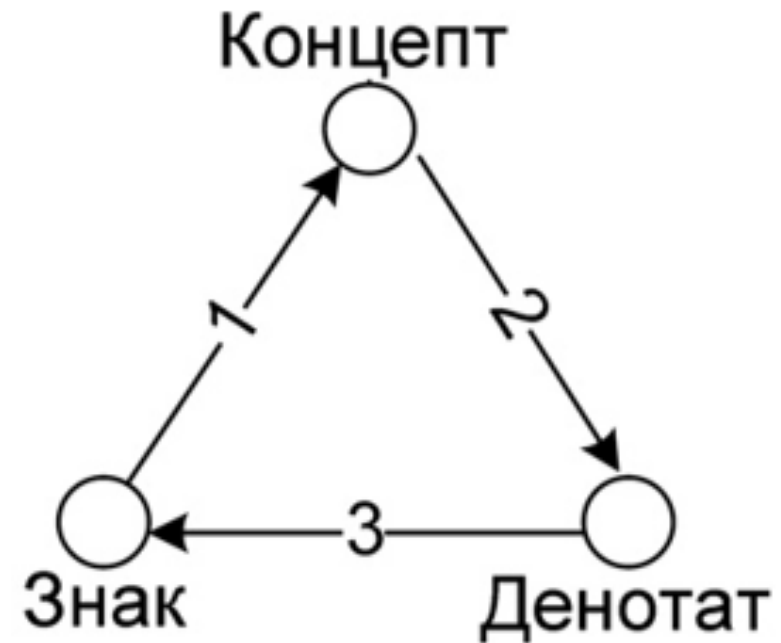
Лекция 1.3 – Оценочные методы принятия решений

Треугольник Фреге - связи (отношения):

связь 1 – активизирует по имени в памяти образ сущности и ее свойства, а в обратном направлении позволяет по описанию сущности определить ее имя;

связь 2 – по представлению о сущности можно найти информацию о ее свойствах или сформировать представление о сущности по совокупности ее свойств;

связь 3 – соединяет представление о денотате с его именем, то есть имя может активизировать представление и наоборот



Треугольник Фреге

Лекция 1.3 – Оценочные методы принятия решений

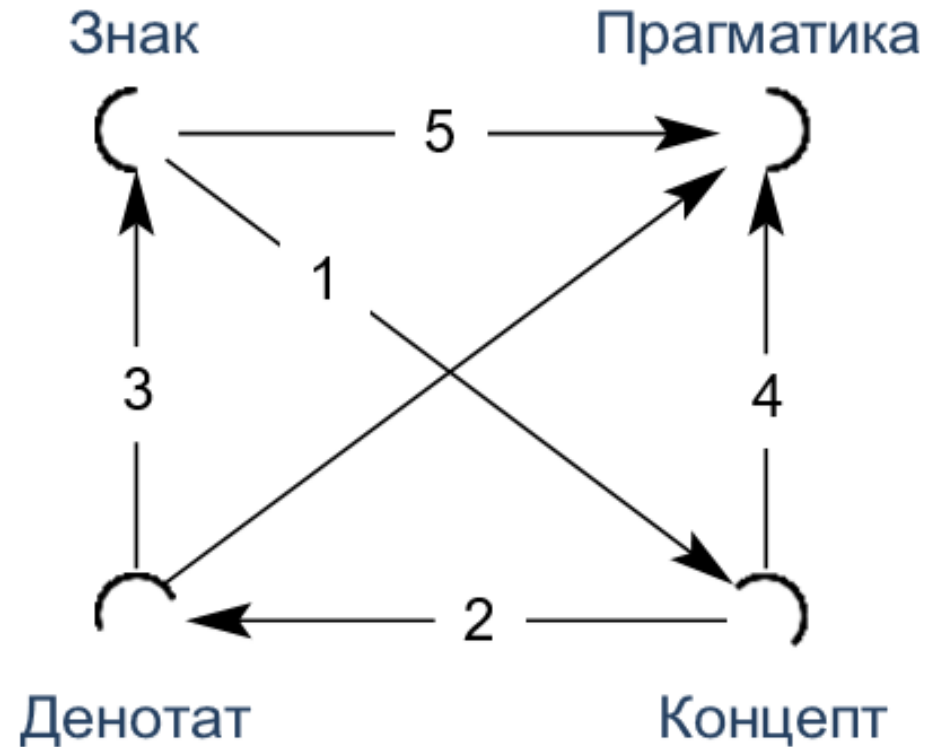
В отношении *элемент – класс* между элементами класса может не быть никакого сходства, кроме их принадлежности одному классу. Определение класса тогда задается путем простого перечисления имен входящих в него элементов ($x \in X$).

В отношении *часть – целое* (Y) наследуется часть свойств сущности, которой приписана роль целого (X) ($Y \subset X$). При этом части могут быть совершенно не похожи друг на друга.

Отношение *род – вид* базируется на наследовании всех свойств X видом Z . Поэтому между видами обязательно сходство.

Лекция 1.3 – Оценочные методы принятия решений

Оценка: кто, как, зачем, почему?



Прямоугольник Поспелова

Лекция 1.3 – Оценочные методы принятия решений

Метод ситуационного управления (1)

Первый принцип акцентирует роль языка в решении задач управления и утверждает, что необходим язык, обеспечивающий такую формализацию проблемы, которая гарантирует достаточную, с точки зрения человека, адекватность описания информационной модели самого объекта, процессов его функционирования и внешней среды, определяющей условия принятия решений в конкретных ситуациях

Лекция 1.3 – Оценочные методы принятия решений

Метод ситуационного управления (2)

Второй принцип базируется на классификации ситуаций. Состоит он в том, что при создании модели управления необходимо разбиение множества всех возможных ситуаций на классы таким образом, чтобы каждому классу можно было сопоставить либо решение по управлению, либо модель, обеспечивающую порождение допустимых рациональных решений. Выбор наилучшего решения из этого множества может происходить на основе либо эвристических процедур, либо известных точных математических моделей.

Предполагается при этом, что мощность множества всех ситуаций существенно больше множества всех средств порождения рациональных решений.

Лекция 1.3 – Оценочные методы принятия решений

Метод ситуационного управления (1)

Третий принцип метода предполагает наличие аппарата, позволяющего строить модель управления объектом на основе обучения этих моделей принятию решений. Обучение должно осуществляться либо экспертом, либо на основе опыта решения задач управления, накапливаемого ситуационной моделью управления в процессе своего функционирования. При этом обучение состоит в формировании самой модели объекта, а затем модели принятия решений по управлению. Этот принцип обеспечивает создание моделей, способных к усовершенствованию функций принятия решений, к адаптации в изменяющихся условиях работы. Это означает также возможность расширения модели, ее функций и ассортимента решаемых задач, реализации принципов и «выращивания» модели для заданных условий.

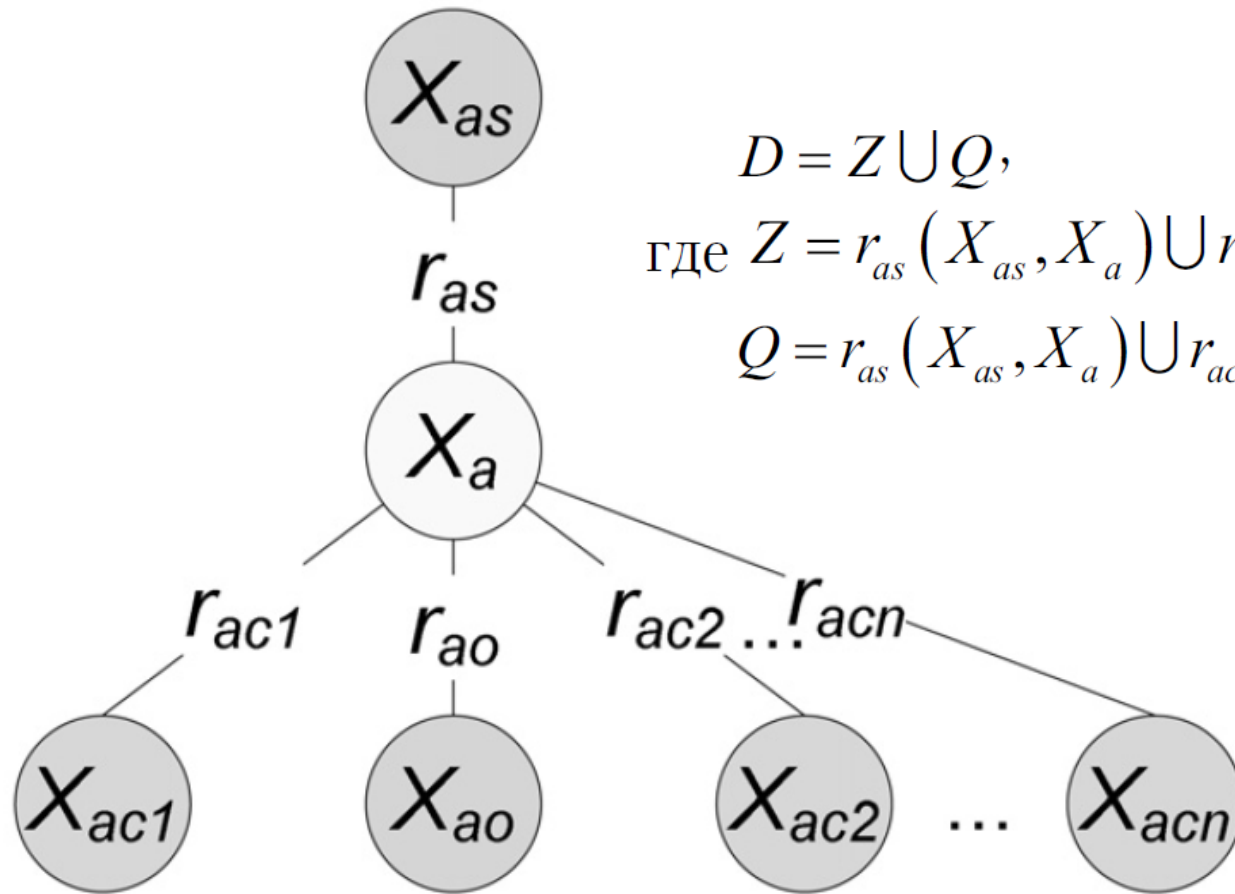
Лекция 1.4 – Количественные методы принятия решений

Структурная схема единичного решения

- 1) в любом решении по управлению само решение отождествляется с действием, у которого есть субъект и объект, описывается оно как императив в виде глагола, задаваемого в инфинитивной форме (сделать, выдать, построить и т.п.);
- 2) решение выступает в роли целеобразующей составляющей задачи по управлению, в этом смысле дерево решений может отождествляться с деревом целей или задач;
- 3) двухуровневый граф со структурой является инвариантом по отношению к произвольной Предметной Области.

Лекция 1.4 – Количественные методы принятия решений

Структурная схема единичного решения



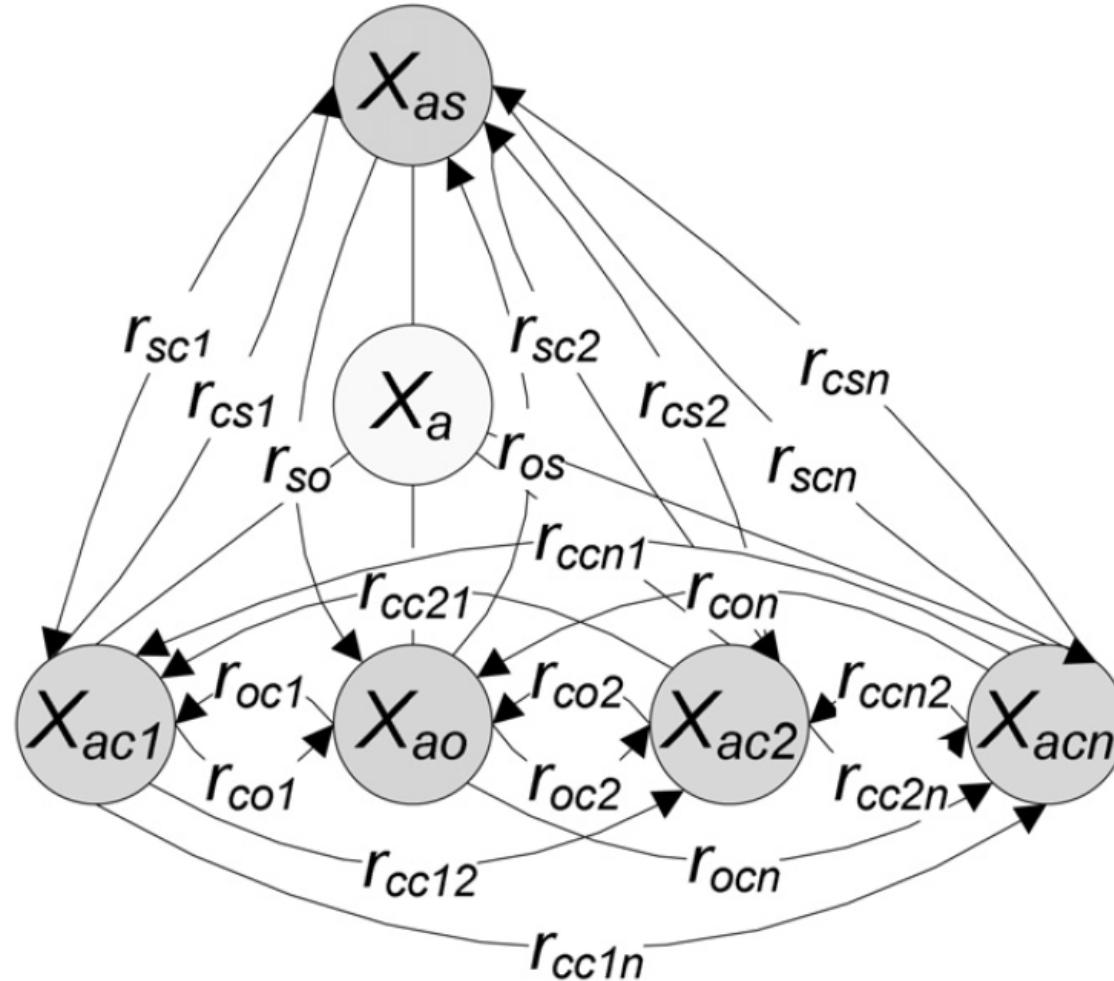
$$D = Z \cup Q,$$

$$\text{где } Z = r_{as}(X_{as}, X_a) \cup r_{ao}(X_a, X_{ao}),$$

$$Q = r_{as}(X_{as}, X_a) \cup r_{ac_1}(X_a, X_{ac_1}) \cup r_{ac_2}(X_a, X_{ac_2}) \cup \dots \cup r_{ac_n}(X_a, X_{ac_n}).$$

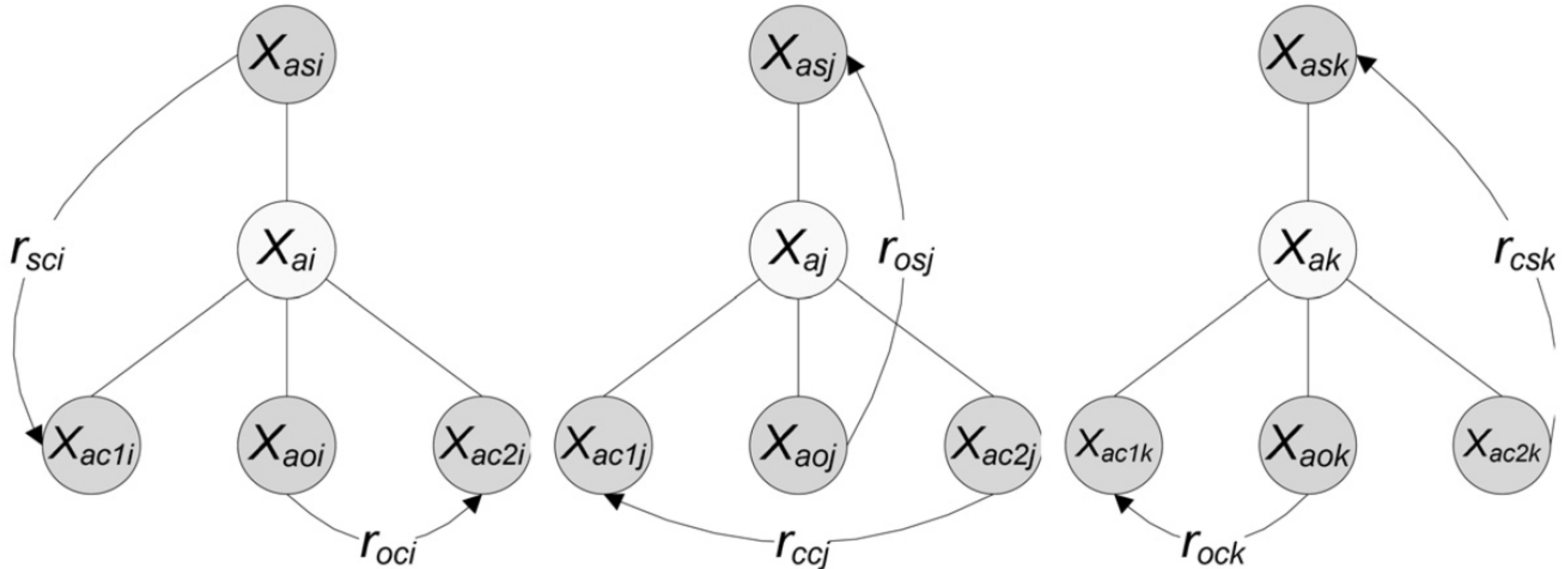
Лекция 1.4 – Количественные методы принятия решений

Семантическая структура единичного решения



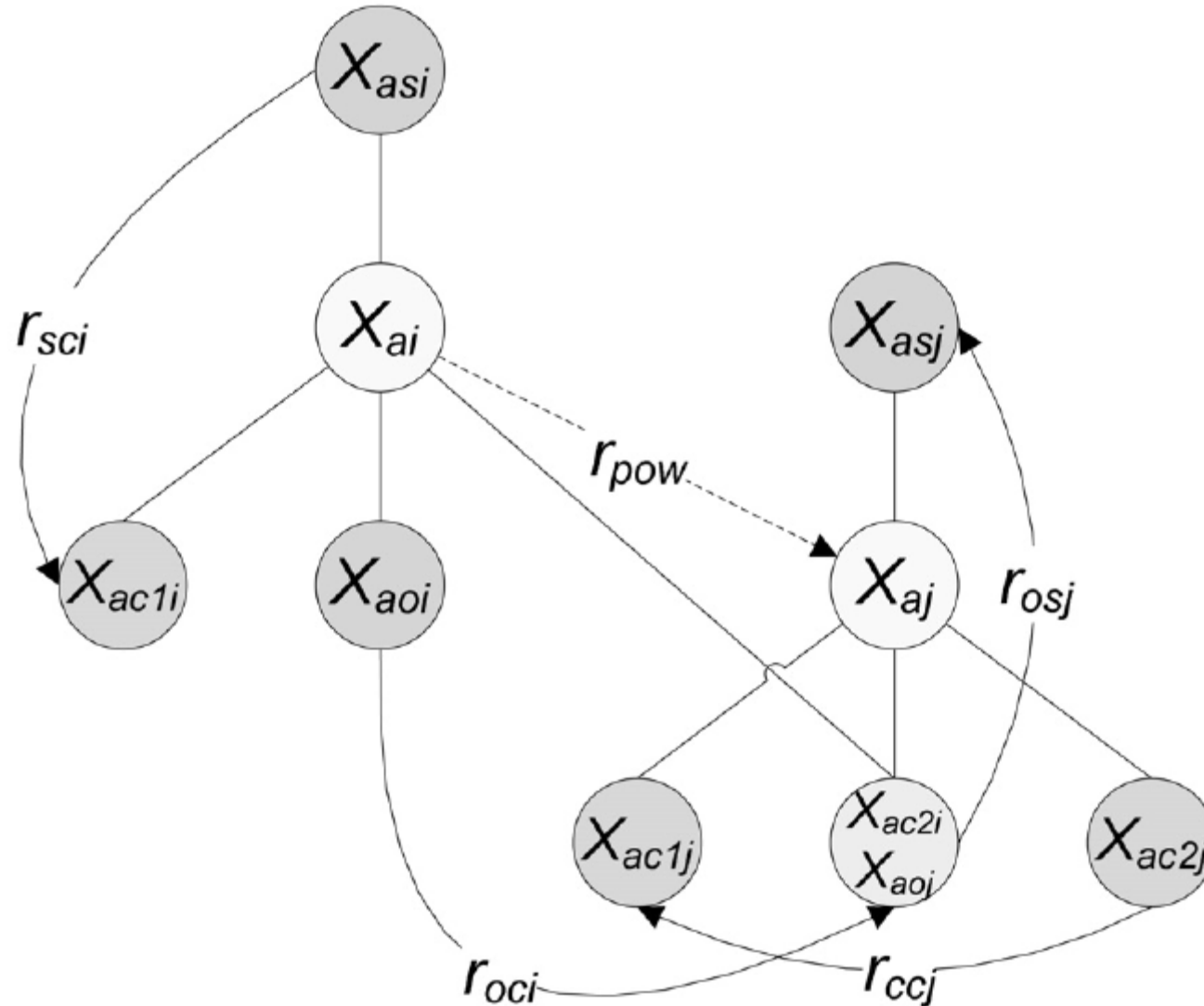
Лекция 1.4 – Количественные методы принятия решений

Структуры с 2 компонентами действия, 4 структурными и 2 семантическими отношениями



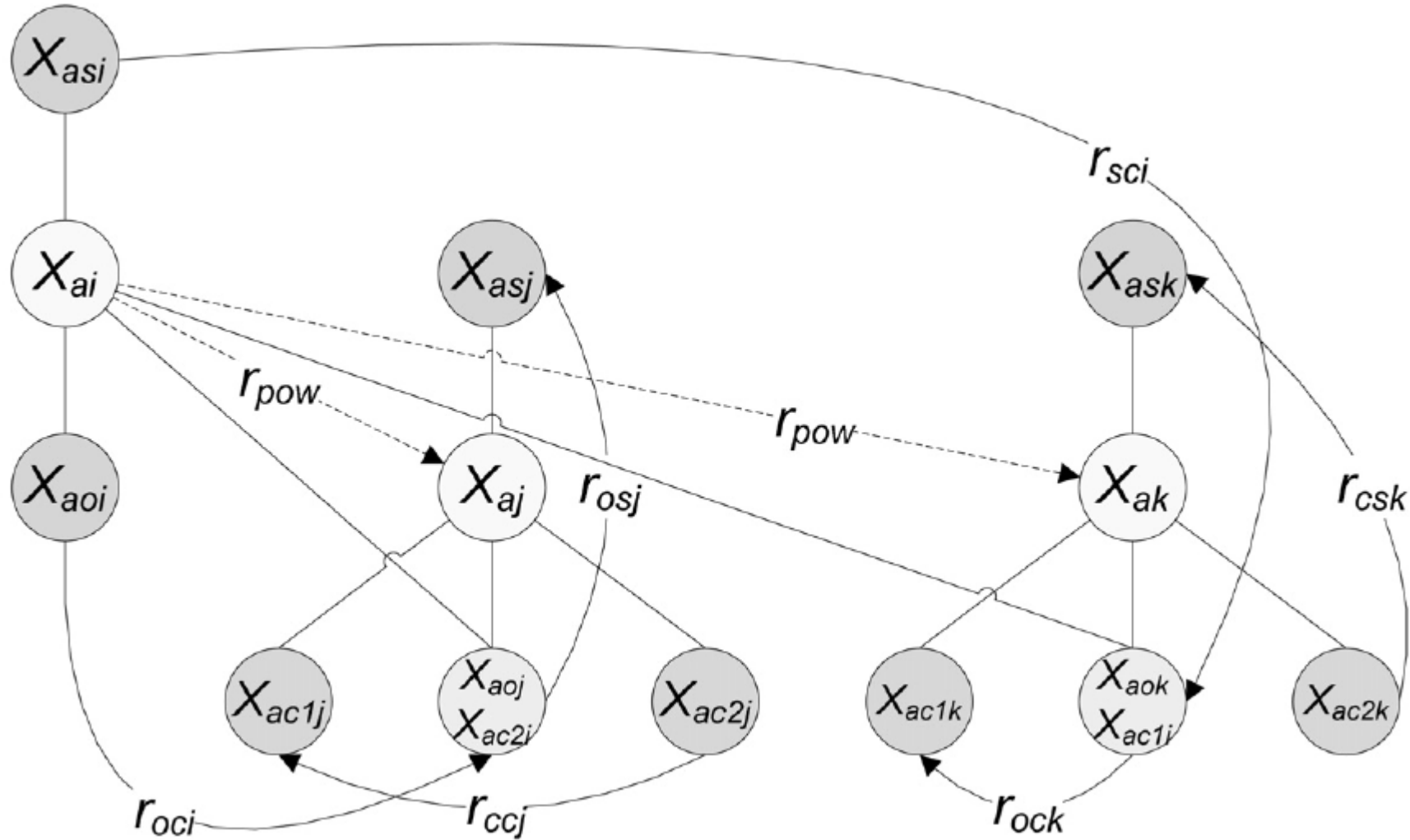
Лекция 1.4 – Количественные методы принятия решений

Проявление неявных отношений «действие-поддействие»



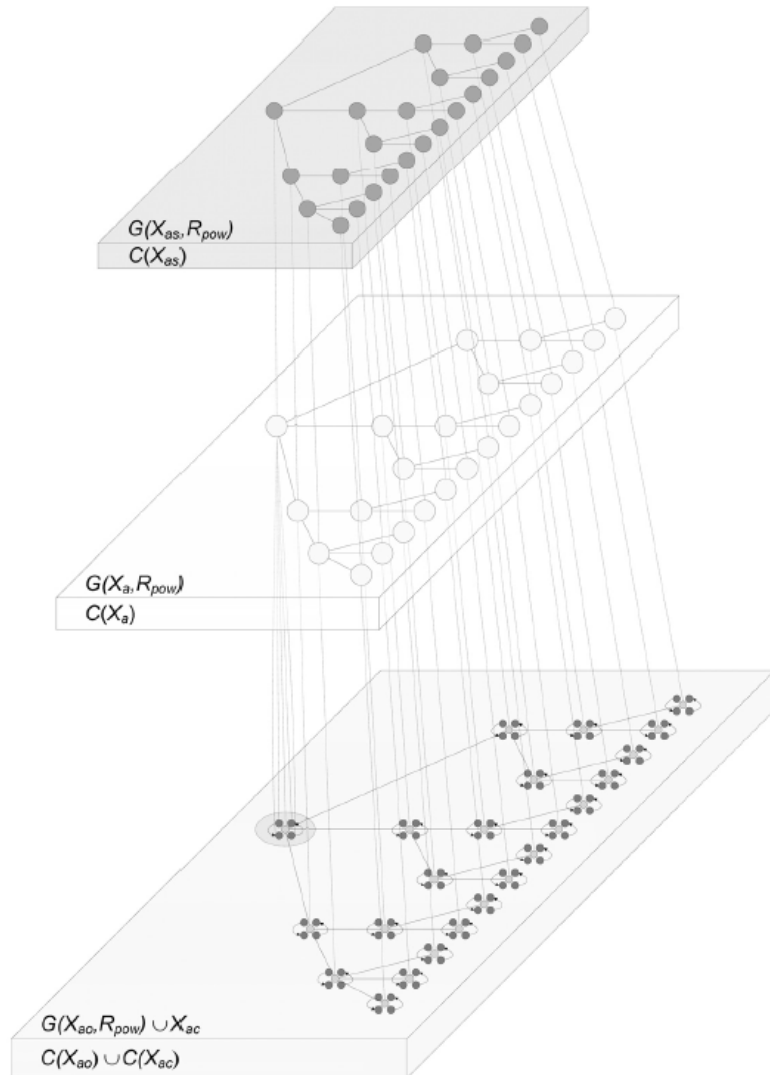
Лекция 1.4 – Количественные методы принятия решений

Проявление неявных отношений «действие-поддействие»



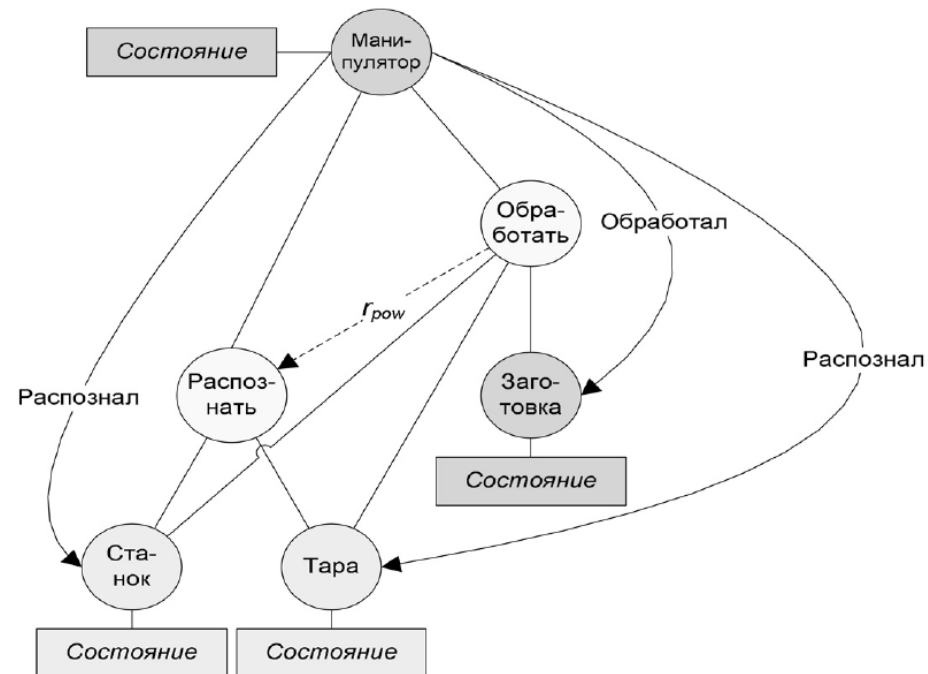
Лекция 1.4 – Количественные методы принятия решений

Концептуальная структура пространства решений



$$G(D) = \bigcup_{i=1}^m C(d_i) \cup G(X_{as}, R_{pow}) \cup G(X_a, R_{pow}) \cup G(X_{ao}, R_{pow})$$

где $\bigcup_{i=1}^m C(d_i)$ – отображение $F_D : S_{d_i} \rightarrow U_{X_{a_j}}$ ($i, j = \overline{1, m}$) или база знаний данной ПрО, остальные члены – иерархии субъектов действий, действий и объектов действий.



Лекция 1.5 – Теоретико-игровые методы принятия решений

Конфликты и принятие решений

Построением математических моделей конфликтных ситуаций и разработкой методов решения возникающих в этих ситуациях задач занимается **теория игр**.

Методы и рекомендации теории игр применимы к многократно повторяющимся конфликтным ситуациям. Если конфликтная ситуация реализуется ограниченное число раз, то рекомендации теории игр теряют смысл.

Игра – это упрощенная математическая модель конфликтной ситуации.

Игра ведется по определенным правилам. Суть игр состоит в том, что каждый участник принимает такое решение, которое, как он полагает, обеспечит ему наилучший исход. Исходом игры называется значение некоторой функции, называемой функцией выигрыша (платежной функцией), которая может задаваться в матричном или аналитическом виде.

Лекция 1.5 – Теоретико-игровые методы принятия решений

Основные понятия

Стратегия – это совокупность правил, однозначно определяющих последовательность действий игрока в каждой конкретной ситуации, складывающейся в процессе игр.

Оптимальной называется стратегия, которая при многократном повторении игры обеспечивает данному игроку максимально возможный средний выигрыш. Количество стратегий у каждого игрока может быть конечным или бесконечным. В зависимости от этого игры подразделяются на конечные и бесконечные.

Игра состоит из отдельных партий.

Партия – это каждый вариант реализации игры.

В партии игроки совершают ходы.

Ход – это выбор и реализация игроком одного из допустимых вариантов поведения

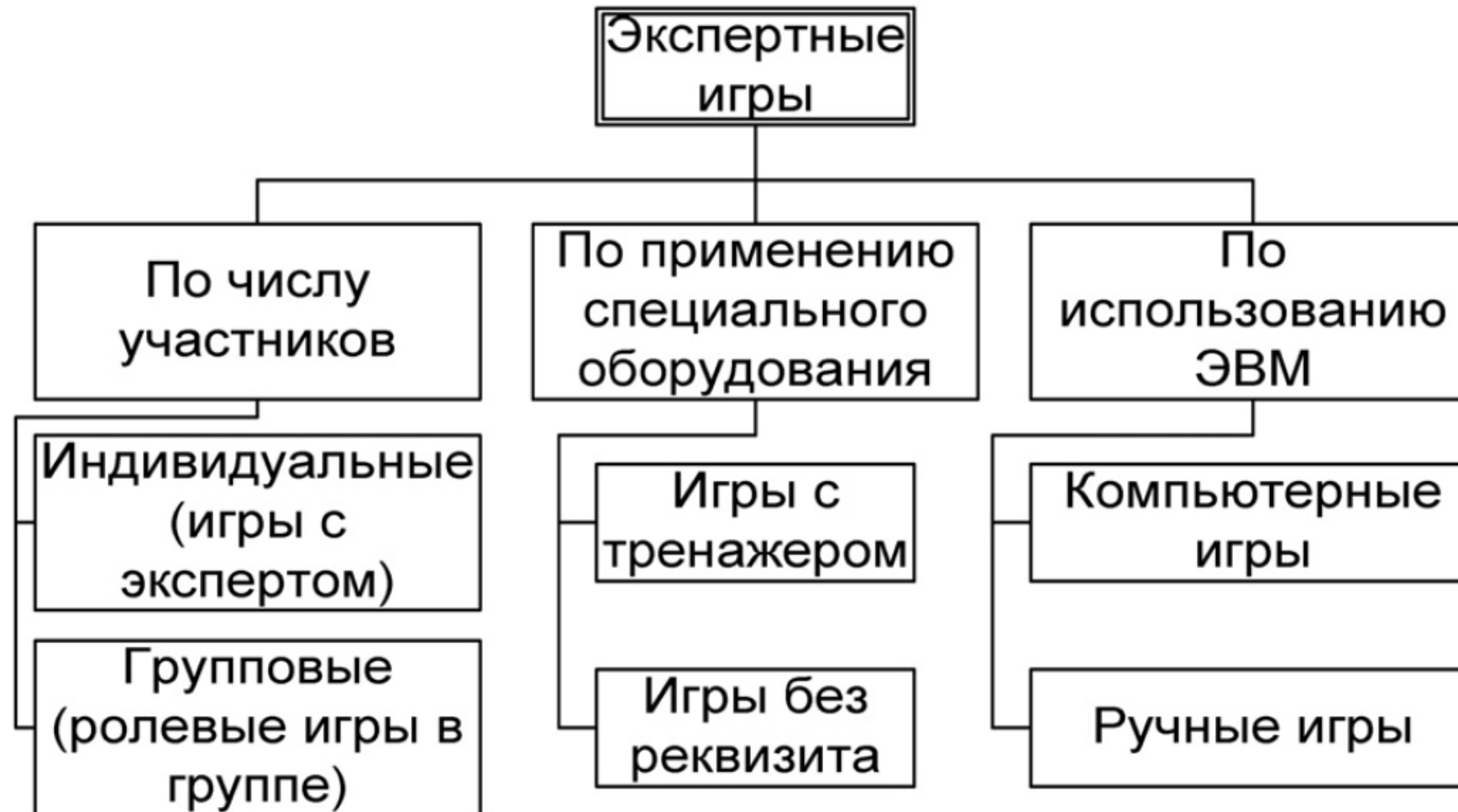
Лекция 1.5 – Теоретико-игровые методы принятия решений

Типы игр в теории игр



Лекция 1.5 – Теоретико-игровые методы принятия решений

Типы игр в экспертных системах



РАЗДЕЛ 2

ПРЕДСТАВЛЕНИЕ ЗНАНИЙ И РАССУЖДЕНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

Одним из достаточно известных средств представления знаний является аппарат формальной логики. В логике разрабатываются методы правильных рассуждений, представляющих собой цепь умозаключений в логически последовательной форме. Рассуждения в ней изучаются с точки зрения формы, а не смысла, и с этой целью в обычных рассуждениях выделяются определенные элементы, которые могут замещаться в них произвольным образом какими-то другими элементами.

ЛОГИЧЕСКАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Формальная система представляет собой совокупность абстрактных объектов (обычно обозначаемых символами) и правил оперирования множеством таких символов в чисто синтаксической трактовке без учета их семантики (или смыслового содержания). Она определена, если:

- задан конечный алфавит α (или конечное множество символов);
- определена процедура построения правильных формул β ;
- выделено некоторое множество формул A , называемых аксиомами;
- задано конечное множество правил вывода P , которые позволяют получать из некоторого конечного множества формул другое множество формул.

Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

ЛОГИЧЕСКАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

ИСЧИСЛЕНИЕ ПРЕДИКАТОВ ПЕРВОГО ПОРЯДКА

Для построения атомов разрешается использовать следующие четыре типа символов:

- ❑ **Индивидуальные символы, или константы;**
- ❑ **Символы предметных переменных.** Это обычно строчные буквы x, y, z, \dots , возможно, с индексами;
- ❑ **Функциональные символы.** Это обычно строчные буквы f, g, h, \dots или осмысленные слова из строчных букв такие, как отец и плюс;
- ❑ **Предикатные символы.** Это обычно прописные буквы P, Q, R, \dots или осмысленные слова из прописных букв.

Термы определяются рекурсивно следующим образом:

- ❑ Константа есть терм.
- ❑ Переменная есть терм.
- ❑ Если f есть n -местный функциональный символ и t_1, \dots, t_n – термы, то $f(t_1, \dots, t_n)$ – терм.
- ❑ Никаких термов, кроме порожденных применением указанных выше правил, нет.

Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

ЛОГИЧЕСКАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Если P – n -местный предикатный символ и t_1, \dots, t_n – термы, то $P(t_1, \dots, t_n)$ – атом (или атомарная формула).

Для построения формул мы можем использовать пять логических связок, приведенных выше, а также два специальных символа \forall и \exists , чтобы характеризовать переменные. Символы \forall и \exists называются соответственно **кванторами (все) общности** и **существования**.

Если x – переменная, то $(\forall x)$ читается как «для всех x », «для каждого x » или «для всякого x », тогда как $(\exists x)$ читается «существует x », «для не-которых x » или «по крайней мере для одного x ».

Правильно построенные формулы или формулы логики первого порядка рекурсивно определяются следующим образом:

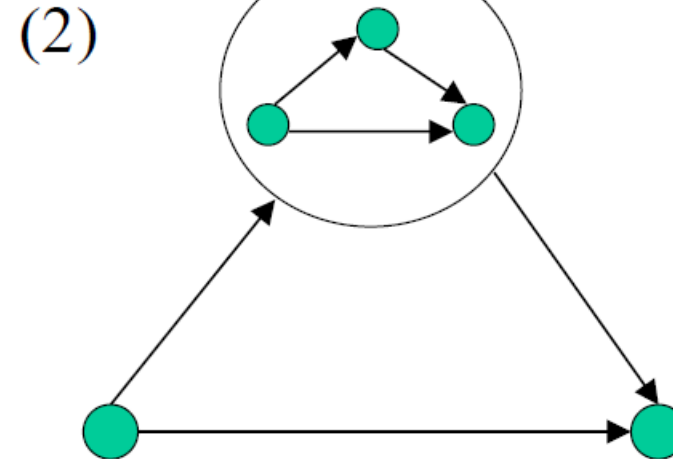
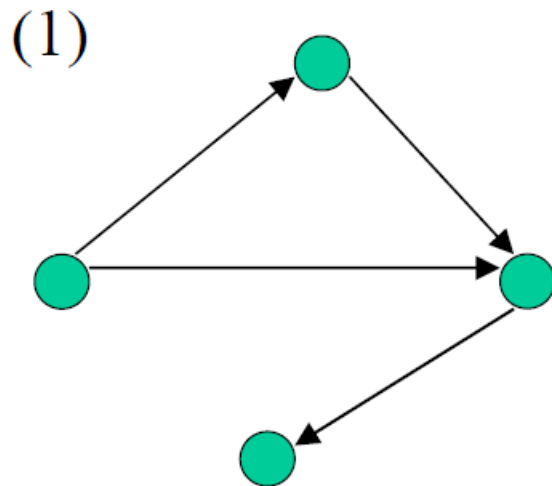
- ❑ Атом есть формула;
- ❑ Если F и G – формулы, то $\neg(F)$, $(F \vee G)$, $(F \wedge G)$, $(F \rightarrow G)$ и $(F \leftrightarrow G)$ – формулы;
- ❑ Если F – формула, а x – свободная переменная в F , то $(\forall x)F$ и $(\exists x)F$ – формулы;
- ❑ Формулы порождаются только конечным числом применений правил/

Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

СЕТЕВАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Семантическая сеть – это ориентированный граф, вершинам которого сопоставляются объекты (понятия, конкретные объекты, события, процессы, явления и т. п.), а дугам – отношения, существующие между объектами.

В зависимости от вида используемых вершин и дуг сети делятся на различные классы. По виду вершин различают (1) простые и (2) иерархические семантические сети:



Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

СЕТЕВАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

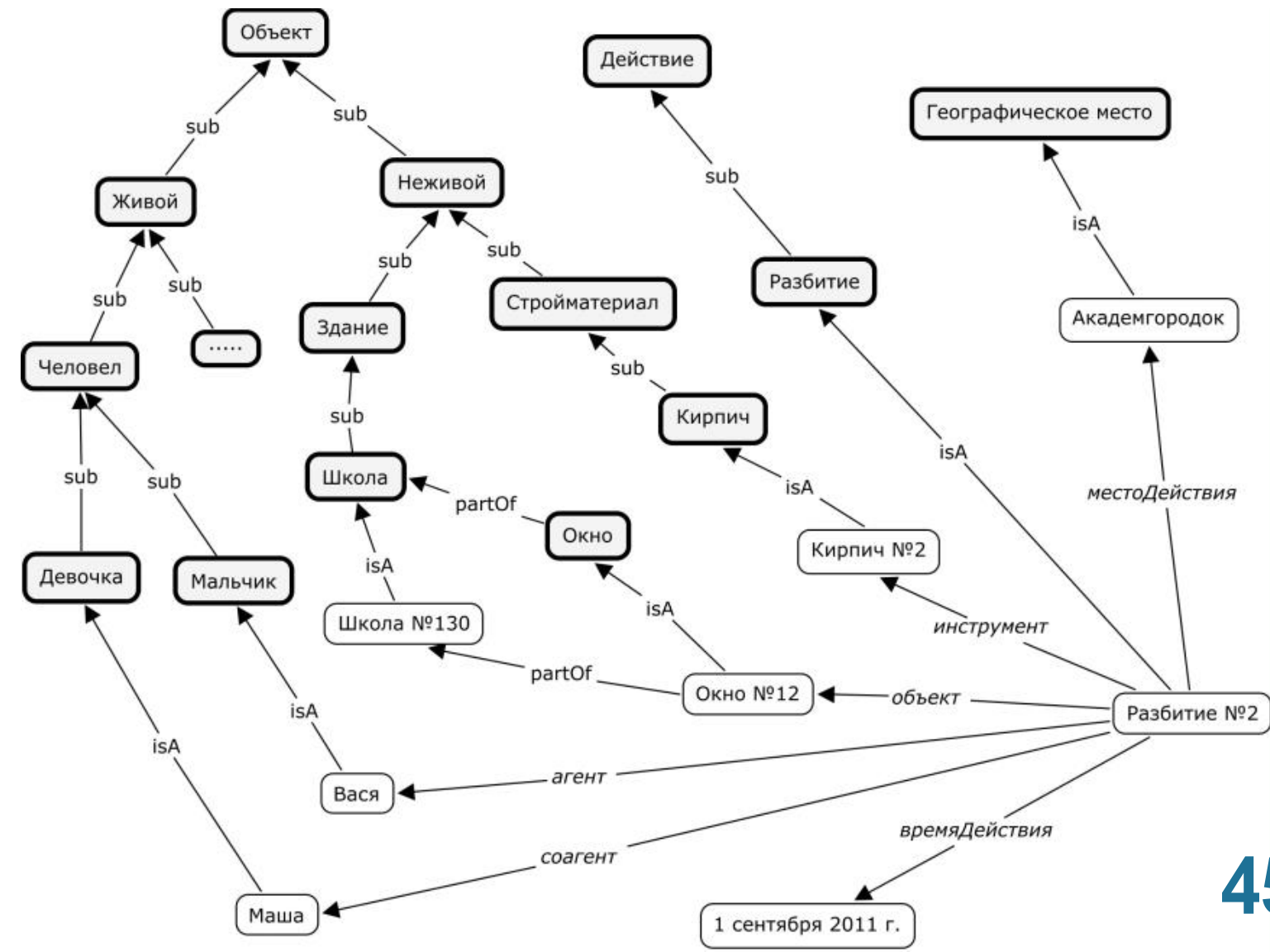
Используемые в **неоднородных семантических сетях** отношения можно разбить на следующие классы:

- ❑ **класс-подкласс (SUB)** – отношение, устанавливаемое между понятиями;
- ❑ **элемент-класс (ISA)** – отношение, устанавливаемое между экземплярами понятий и понятиями;
- ❑ **часть-целое (part_of)**, или **целое-часть** – отношение включения одного объекта в другой;
- ❑ **атрибутивные отношения** или **отношения типа свойство-значение** (“цвет”, “вес”, “рост” и т. п.);
- ❑ **темпоральные отношения** (“раньше”, “позже”, “одновременно” и т. п.);
- ❑ **логические отношения** (“и”, “или”, “не”, “следование”);
- ❑ **глубинно-падежные семантические отношения Филмора** – отношения, пришедшие из лингвистики и служащие для выражения в предложении глубинных семантических отношений между группой существительного и действием

Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

СЕТЕВАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Содержательный пример семантической сети:



Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

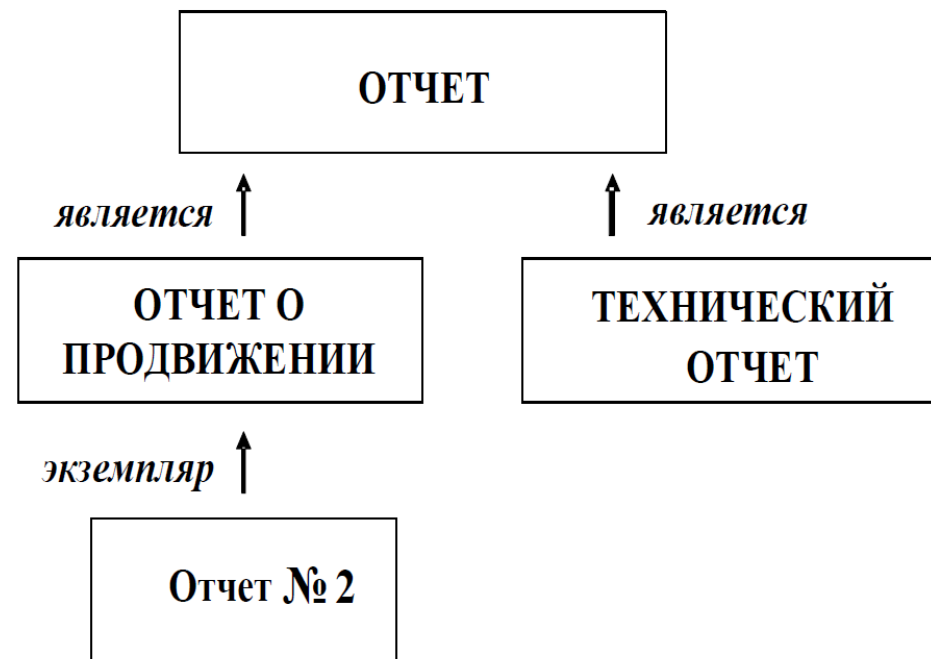
ФРЕЙМОВАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Фрейм может быть представлен в виде сети из узлов и отношений, также может быть представлен из имени и отдельных информационных единиц, называемых слотами:

$$F: [<r1, v1>, <r2, v2>, \dots, <rn, vn>]$$

В отличие от семантических сетей во фрейм-представлении фиксируется жесткая структура информационных единиц, которая задается в фрейм-прототипе, или протофрейме.

Иллюстрация работы системы, основанной на фреймах:



Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

ФРЕЙМОВАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

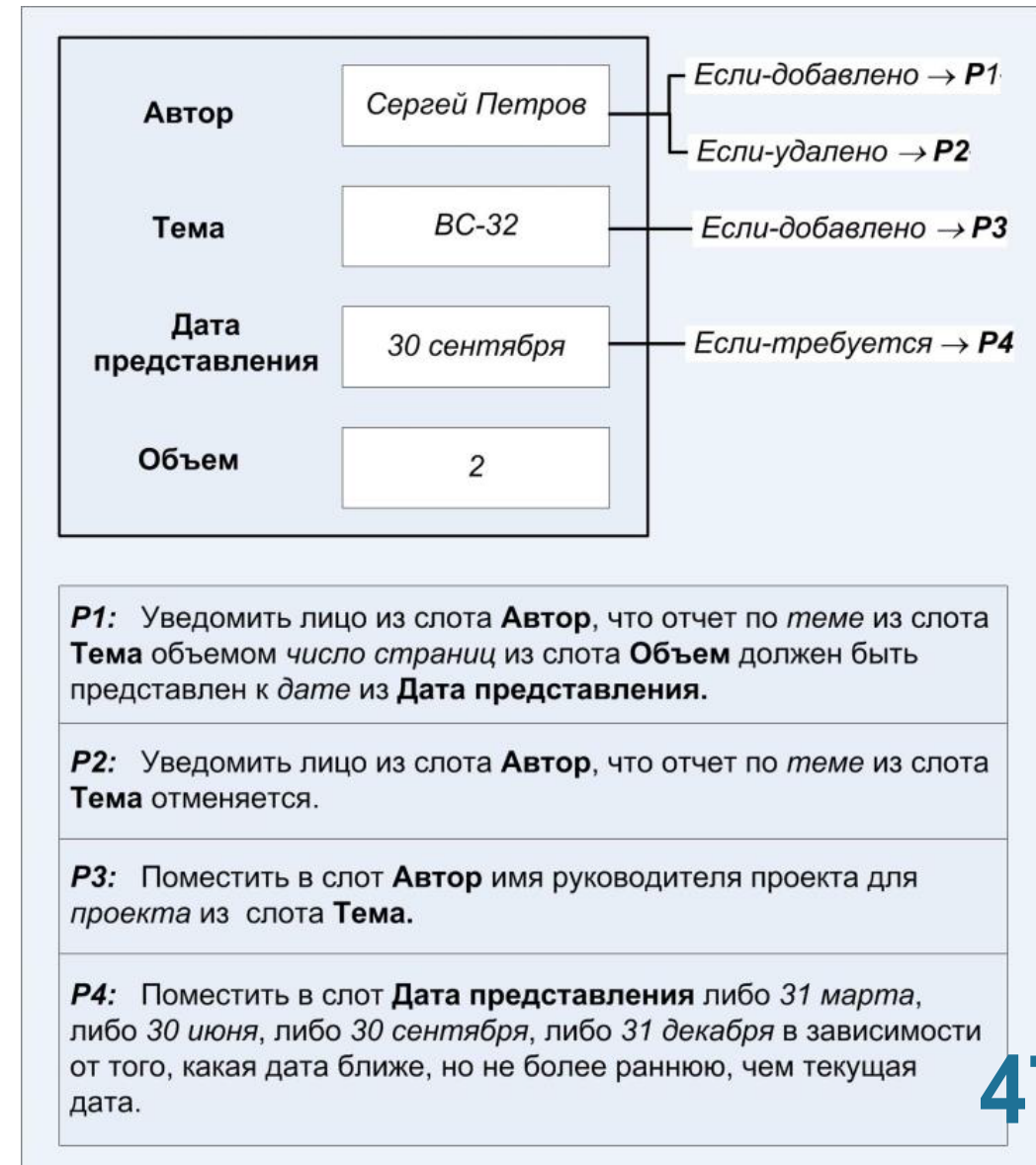
Фрейма отчета о продвижении:

Основным недостатком фрейм-представления является отсутствие общей теории. В нем много эвристик и вариантов решения.

К главным достоинствам фрейм-представления можно отнести иерархичность описания понятий предметной области и достаточно естественное объединение в рамках одного средства как декларативного, так и процедурного компонентов представления знаний.

Следует заметить, что фрейм-представление является не конкретным языком представления знаний, а идеологической концепцией, реализуемой в каждом конкретном языке.

Наиболее известными языками представления знаний на основе фреймов являются **KRL (knowledge representation language)** и **FRL (frame representation language)**.



Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

ПРОДУКЦИОННАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Продукционная модель предполагает такой способ организации вычислительного процесса, при котором программа преобразования некоторой информационной структуры S задается в виде системы правил вида: $УСЛОВИЕ \rightarrow ДЕЙСТВИЕ$, где $УСЛОВИЕ$ описывает определенные требования к текущему состоянию структуры S , а $ДЕЙСТВИЕ$ содержит набор тех операций над S , которые надо выполнить, если S удовлетворяет этим требованиям.

Формальными системами, удовлетворяющими требованиям продукционной модели, являются системы подстановок и формальные грамматики.

Системы подстановок представляют собой правила преобразования цепочек (слов), заданных в некотором алфавите. К таким системам относятся системы productions Поста (именно этим системам мы обязаны появлению термина «система productions», который получил со временем более широкое употребление) и нормальные алгоритмы Маркова.

Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

ПРОДУКЦИОННАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Системы Поста определяются алфавитом S и набором правил-продукций вида:

$$a_i W \rightarrow W b_i \quad (i = 1, \dots, m)$$

Нормальные алгоритмы Маркова определяются алфавитом S и последовательностью подстановок вида:

$$\begin{aligned} a_i &\rightarrow b_i \\ a_i &\rightarrow \cdot b_i \end{aligned}$$

Структура программной системы продукции (СП):

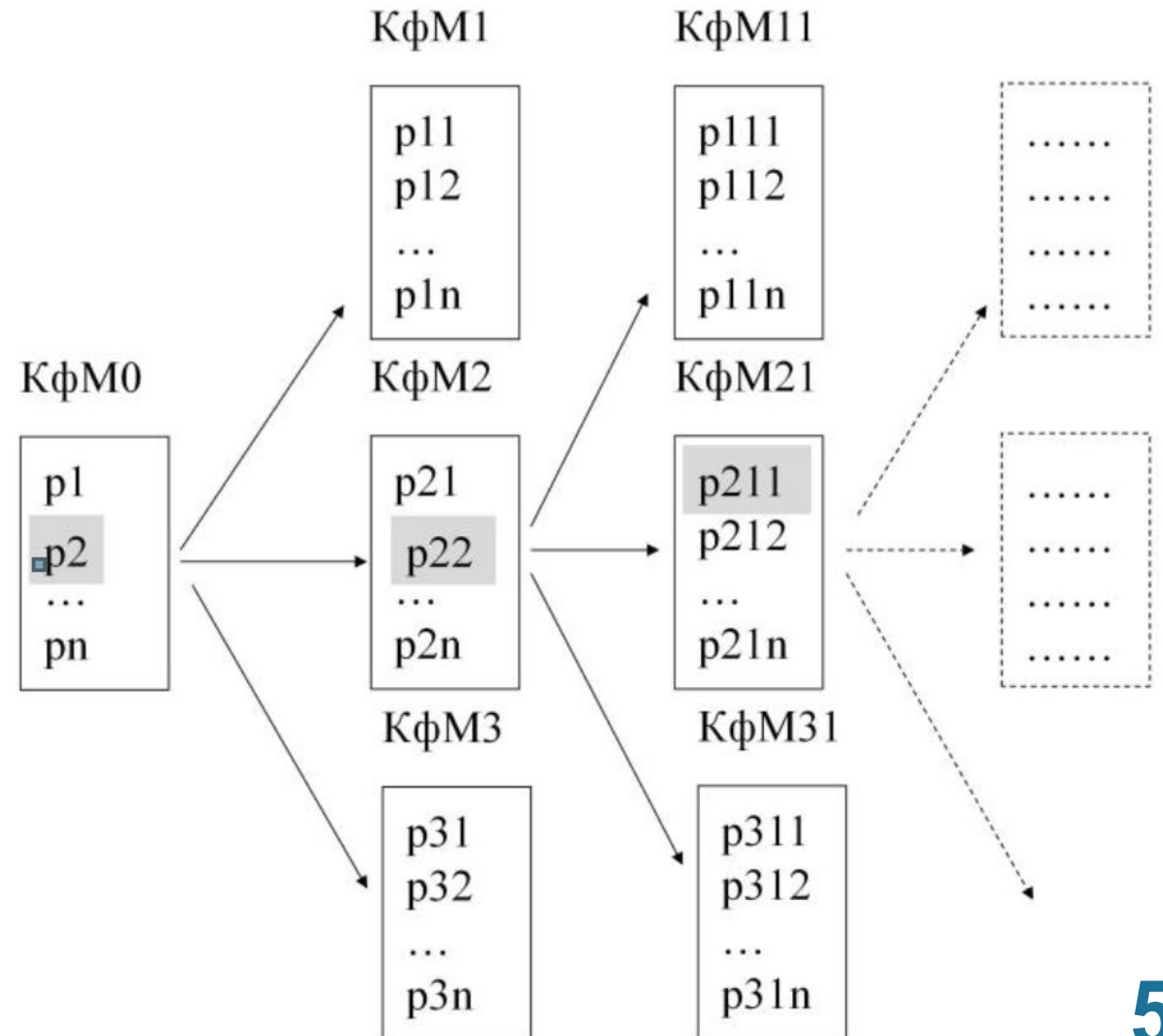


Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

ПРОДУКЦИОННАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Основные способы выбора продукций из конфликтного множества:

- ❑ случайный выбор;
- ❑ выбор по заранее оговоренному критерию, не меняемому в ходе вычислений (например, первой применяется продукция с самыми жесткими требованиями – продукция с самым длинным списком условий, либо на продукциях задан полный порядок или иерархия, при этом первой применяется самая «старшая» продукция и т. п.);
- ❑ выбор по критерию, формируемому динамически по ходу процесса, например, приписыванием правилам и/или компонентам базы динамически вычисляемых весов (приоритетов) (в этом случае первой для исполнения выбирается продукция с наивысшим приоритетом, или продукция, условия которой удовлетворяются на данных, имеющих максимальный приоритет).



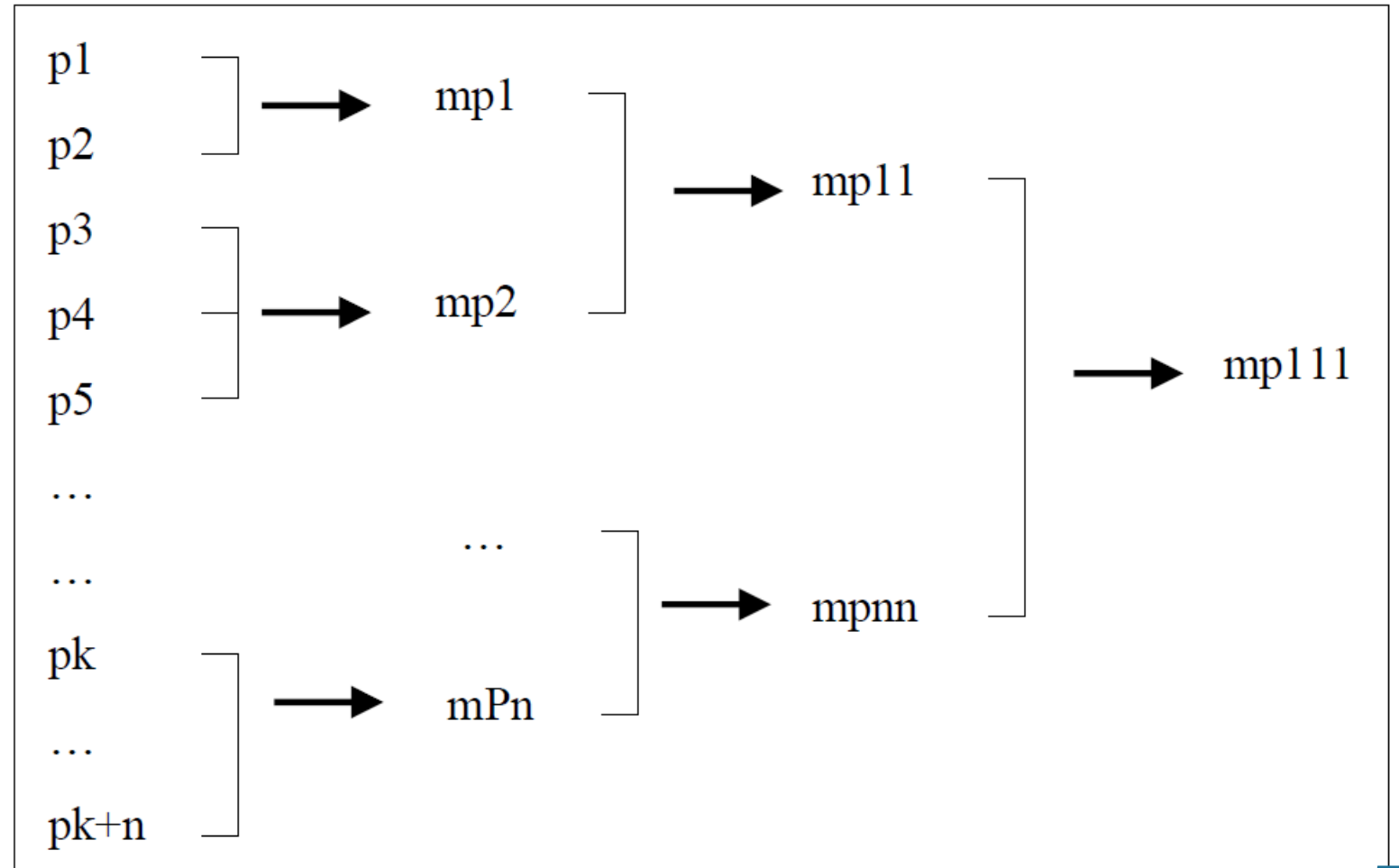
Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

ПРОДУКЦИОННАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Продукционная модели используются в следующих областях:

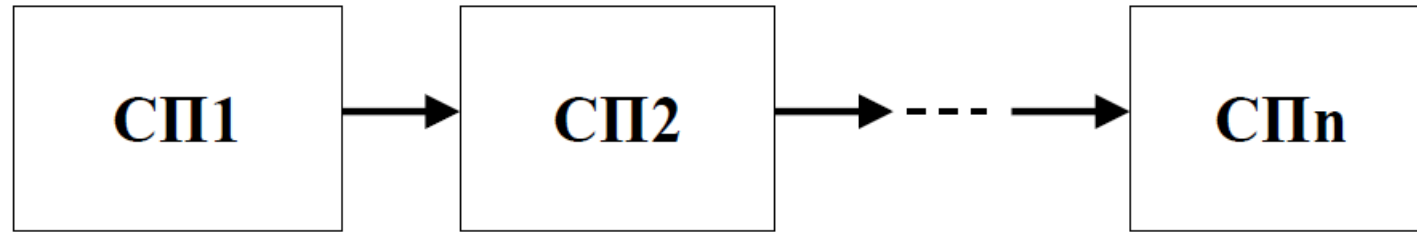
- ❑ Построение компиляторов;
- ❑ Автоматическая обработка текстов;
Распознавание и синтез речи;
- ❑ Экспертные системы.

Структура иерархической СП:

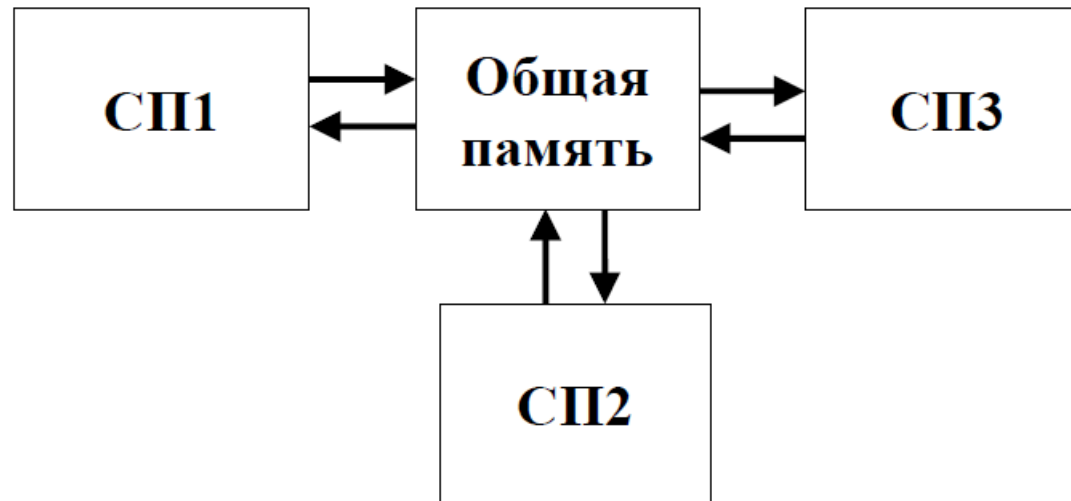


Лекция 2.1 - Логическая, сетевая, продукционная и фреймовая модели представления знаний

ПРОДУКЦИОННАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ



Структура последовательной СП



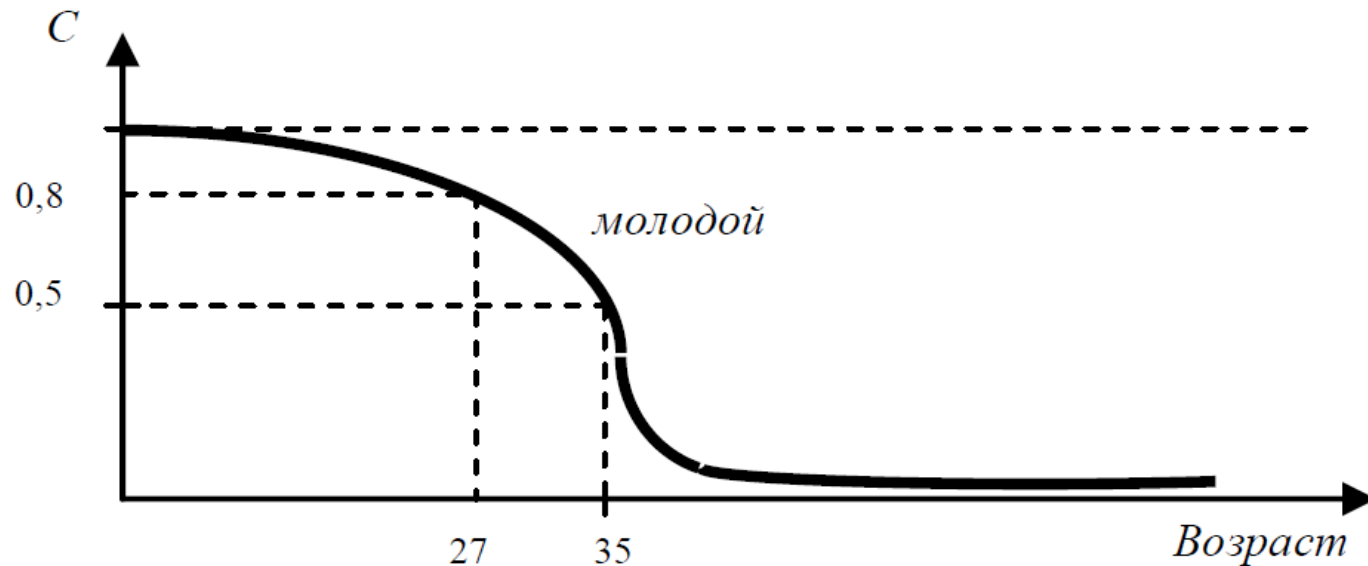
Структура параллельно-последовательной СП

Лекция 2.2 - Представление нечетких знаний

Часто, при представлении знаний о сложных предметных областях приходится сталкиваться с их неполнотой, неточностью, неоднозначностью и нечеткостью. Нечеткость связана с отсутствием точных границ области определений и свойственна большинству понятий. Эта нечеткость границ приводит к тому, что не всегда оказывается возможным решить вопрос о соответствии данного объекта данному понятию по принципу да/нет. Часто можно говорить только о степени соотнесенности одного другому, оценивая ее, например, в интервале от 1 (определенное да) до 0 (определенное нет).

ПОНЯТИЕ ЛИНГВИСТИЧЕСКОЙ ПЕРЕМЕННОЙ

Лингвистическая переменная (ЛП) – это переменная, значениями которой являются слова или выражения естественного (иногда искусственного) языка.



Лекция 2.2 - Представление нечетких знаний

НЕЧЕТКИЕ МНОЖЕСТВА

Нечеткое множество (НМ):

$$A = \{(x, \mu_A(x))\}$$

$$\mu_A : X \rightarrow [0, 1]$$

Универсальным множеством (УМ) X нечеткого множества A называется область определения функции принадлежности μ_A .

Носителем НМ A называется множество таких точек в X , для которых:

$$\mu_A(x) > 0$$

Высотой НМ A называется величина:

$$\sup_X \mu_A(x)$$

Точкой перехода НМ A называется такой элемент множества X , степень принадлежности которого множеству A равна 0,5.

Лекция 2.2 - Представление нечетких знаний

НЕЧЕТКИЕ ОТНОШЕНИЯ

При выполнении нечетких выводов необходимо знать нечеткие отношения. Если X – декартово произведение n универсальных множеств X_1, X_2, \dots, X_n , то n -арное **нечеткое отношение (НО)** R в X определяется как нечеткое подмножество X :

$$R = \int_{X_1 \times X_2 \times \dots \times X_n} \mu_R(x_1, x_2, \dots, x_n) | (x_1, x_2, \dots, x_n)$$

где μ_R – функция принадлежности НМ R .

Распространенными примерами нечетких отношений являются «*много больше чем*», «*имеет сходство*», «*близко к*» и т. д.

Например, если $X_1 = X_2 = (-\infty + \infty)$, то отношение «*близко к*» можно определить следующим образом:

$$\text{«близко к»} = \int_{X_1 \times X_2} e^{-\alpha * |x_1 - x_2|} | (x_1, x_2)$$

где α - коэффициент масштабирования

Лекция 2.2 - Представление нечетких знаний

ИСПОЛЬЗОВАНИЕ НЕЧЕТКОЙ ЛОГИКИ В СИСТЕМАХ, ОСНОВАННЫХ НА ЗНАНИЯХ

Особенности нечеткой логики:

В нечеткой логике определены эквиваленты операций И, ИЛИ и НЕ:

- $p1$ И $p2 = \min(p1, p2)$ (т. е. меньшее);
- $p1$ ИЛИ $p2 = \max(p1, p2)$ (т. е. большее);
- НЕ $p1 = 1 - p1$ (т. е. «обратное значение»).

Наибольшее распространение получили следующие **функции принадлежности**:

- треугольная;
- трапецеидальная;
- гауссова (нормальное распределение).

Лекция 2.2 - Представление нечетких знаний

ИСПОЛЬЗОВАНИЕ НЕЧЕТКОЙ ЛОГИКИ В СИСТЕМАХ, ОСНОВАННЫХ НА ЗНАНИЯХ

Схема Шортлиффа:

Э. Шортлифф (E. Shortliffe) разработал схему, основанную на так называемых коэффициентах уверенности, которые он ввел для измерения степени доверия к любому данному заключению, являющемуся результатом полученных к этому моменту свидетельств.

Коэффициент уверенности – это разность между двумя мерами:

$$КУ [h : e] = МД [h : e] - МНД [h : e]$$

Формула уточнения:

$$МД [h : e1, e2] = МД [h : e1] + МД [h : e2] * (1 - МД [h : e1]).$$

Данная формула имеет два важных свойства:

1. Она **симметрична** в том смысле, что порядок $e1$ и $e2$ не существенен.
2. По мере накопления подкрепляющих свидетельств $МД$ (или $МНД$) движется к определенности.

Каждое правило снабжается **коэффициентом ослабления** (числом от 0 до 1), показывающим **надежность правила**.

Для обеспечения возможности исключения из вывода заключений, имеющих низкую достоверность, введен **порог уверенности (ПУ)** – число от 0 до 1.

Лекция 2.2 - Представление нечетких знаний

МЕТОДАМИ ТЕОРИИ ПРИБЛИЖЕННЫХ МНОЖЕСТВ

Одной из метрических характеристик приближенного множества является функция принадлежности приближенного множества

$$\mu_A(x) = \frac{\text{card}([x]_B \cap Z)}{\text{card}([x]_b)}$$

где $Z \subseteq U$, $\text{card}(\bullet)$ – это мощность множества.

Агрегации характеристик атрибутов использует следующие отношения:

$$R_{\min}(x, y) = \frac{\sum_{a \in V_a} \min\{\mu_x(a), \mu_y(a)\}}{\sum_{a \in V_a} \mu_y(a)}$$

$$R_{\max}(x, y) = \frac{\sum_{a \in V_a} \max\{\mu_x(a), \mu_y(a)\}}{\sum_{a \in V_a} \mu_y(a)}$$

$$R_{\text{avg}}(x, y) = \frac{\sum_{a \in V_a} \text{avg}\{\mu_x(a), \mu_y(a)\}}{\sum_{a \in V_a} \mu_y(a)}$$

Лекция 2.2 - Представление нечетких знаний

МЕТОДЫ ТЕОРИИ ПРИБЛИЖЕННЫХ МНОЖЕСТВ

Одной из метрических характеристик приближенного множества является функция принадлежности приближенного множества

$$\mu_A(x) = \frac{\text{card}([x]_B \cap Z)}{\text{card}([x]_b)}$$

где $Z \subseteq U$, $\text{card}(\bullet)$ – это мощность множества.

Агрегации характеристик атрибутов использует следующие отношения:

$$R_{\min}(x, y) = \frac{\sum_{a \in V_a} \min\{\mu_x(a), \mu_y(a)\}}{\sum_{a \in V_a} \mu_y(a)}$$

$$R_{\max}(x, y) = \frac{\sum_{a \in V_a} \max\{\mu_x(a), \mu_y(a)\}}{\sum_{a \in V_a} \mu_y(a)}$$

$$R_{\text{avg}}(x, y) = \frac{\sum_{a \in V_a} \text{avg}\{\mu_x(a), \mu_y(a)\}}{\sum_{a \in V_a} \mu_y(a)}$$

Лекция 2.3 - Онтологические модели представления знаний

ОНТОЛОГИИ

Основная цель создания онтологий заключается в обеспечении поддержки деятельности по накоплению, совместному использованию и повторному использованию знаний. Чтобы сделать такую деятельность возможной, создаются стандарты, регламентирующие структуру и процесс разработки онтологий, включая языки описания онтологий.

Термин «онтология» имеет всего два основных значения:

1. **Онтология** – это раздел философии, учение о бытии (в отличие от гносеологии – учения о познании), в котором исследуются всеобщие основы, принципы бытия, его структура и закономерности.
2. **Онтология** – это некий инженерный артефакт, структура, описывающая некоторую реальность (или систему) с помощью слов заданного словаря и множества допущений на значения этих слов.

Под формальной моделью онтологии O понимают тройку вида:

$$O = \langle C, R, F \rangle$$

где C – конечное множество понятий (концептов) предметной области (ПО), которую определяет онтология O ; R – конечное множество отношений между понятиями ПО; F – конечное множество функций интерпретации (аксиоматизация), заданных на понятиях и/или отношениях онтологии O .

Лекция 2.3 - Онтологические модели представления знаний

ОНТОЛОГИИ

Основными компонентами современных онтологий являются:

- ❑ **Классы (classes)**. Обычно организованы в таксономии;
- ❑ **Отношения (relations)**. Представляют тип связей между концептами предметной области;
- ❑ **Функции (functions)**. Специальный случай отношений, в которых n-й элемент определяется по значениям (n-1) предшествующих элементов: $F: C_1 \times C_2 \times \dots \times C_{n-1} \Rightarrow C_n$;
- ❑ **Аксиомы (axioms)**. Моделируют предложения, которые всегда истинны;
- ❑ **Экземпляры (instances)**. Представляют конкретные объекты реального или абстрактного мира.

Для таксономических отношений (*subclass_of*) имеют место следующие свойства:

- ❑ Транзитивность: $A \text{ subclass_of } B, B \text{ subclass_of } C \Rightarrow A \text{ subclass_of } C$;
- ❑ Наследование свойств (property): $S \text{ property } A, B \text{ subclass_of } A \Rightarrow S \text{ property } (B)$.

Лекция 2.3 - Онтологические модели представления знаний

ОНТОЛОГИИ

Наиболее важной и в теоретическом, и практическом плане представляется классификация онтологий по назначению (или по цели создания):

❑ **Онтологии верхнего уровня (*top-level ontology*).**

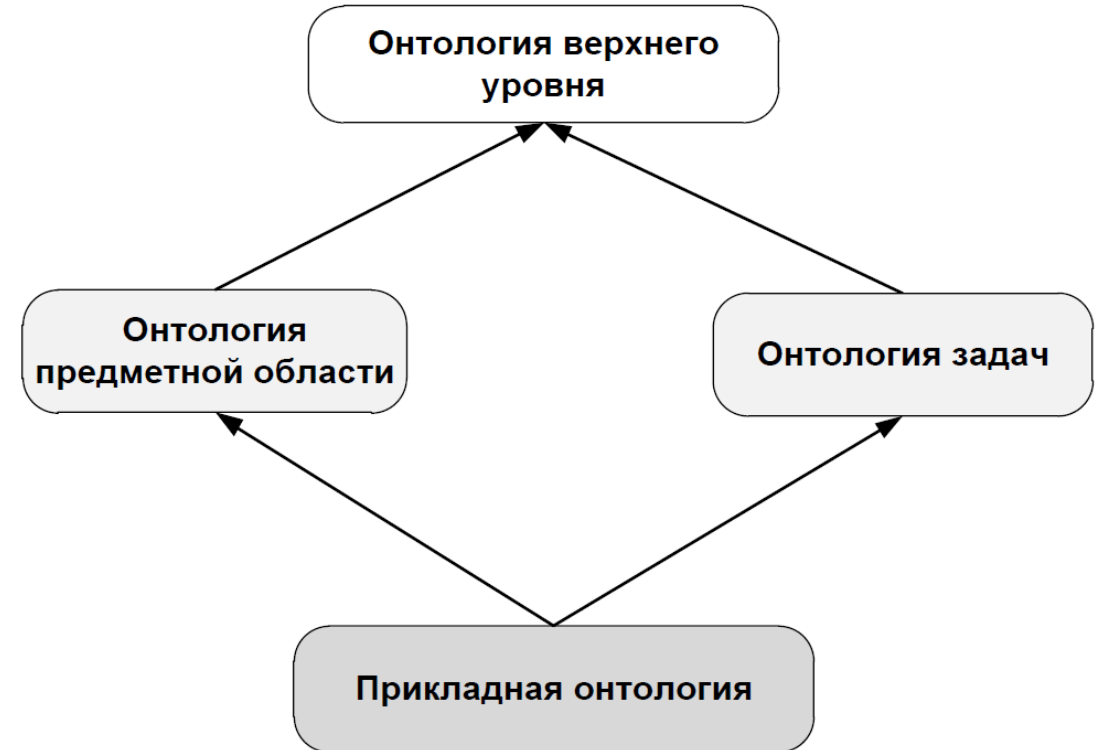
Включают самые общие понятия, которые не зависят от конкретных предметных областей и задач (являются общими для них). Такими понятиями могут быть «время», «пространство», «объект», «событие» и т. д.;

❑ **Онтологии предметных областей (*domain ontology*).**

Описывают понятия и отношения, характерные для конкретных предметных областей (например, медицины, археологии, энергетики);

❑ **Онтологии задач (*task ontology*).** Включают понятия и отношения, описывающие конкретную задачу или деятельность (например, диагностику или обучение);

❑ **Онтология приложения (*application ontology*).** Объединяет в себе онтологию задач и онтологию предметной области для того, чтобы специализировать входящие в них понятия для применения в конкретном приложении.



Лекция 2.3 - Онтологические модели представления знаний

ПРИМЕНЕНИЕ ОНТОЛОГИЙ

Онтологии широко применяются в таких **классах систем**, основанных на знаниях, как:

- Экспертные системы;
- Системы поддержки принятия решений;
- Порталы научных знаний;
- Обучающие и советующие системы;
- Системы управления знаниями предприятия;
- Лингвистические процессоры.

Наиболее **важными задачами**, решаемыми с помощью онтологий, являются:

- Семантическое (онтологическое) описание Интернет-ресурсов, доступное поисковым агентам (подход Semantic Web);
- Информационный поиск (расширение и специализация запросов на основе онтологии для повышения релевантности и полноты ответов);
- Интеграция разнородных источников данных (использование онтологии на концептуальном уровне и при поиске);
- Обработка текстов на естественном языке (модель предметной области задается в виде онтологии).

Лекция 2.3 - Онтологические модели представления знаний

ПРИМЕНЕНИЕ ОНТОЛОГИЙ

При разработке интеллектуальных систем онтология может использоваться для:

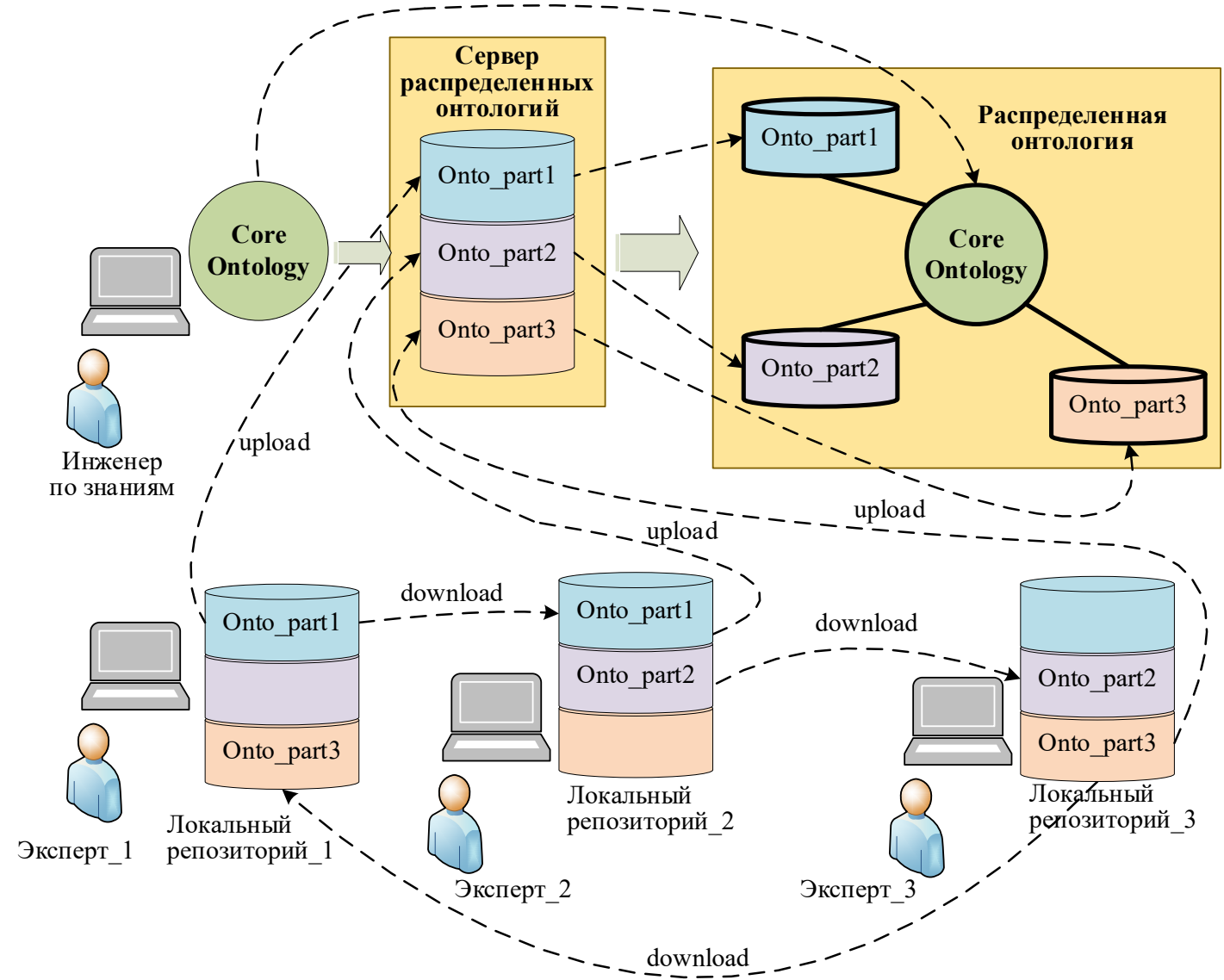
- формирования и фиксации общего разделяемого всеми экспертами знания о предметной области;
- явной концептуализации предметной области, позволяющей описывать семантику данных;
- обеспечения возможности переиспользования знаний;
- описания функциональности системы (типов решаемых задач).

В процессе функционирования интеллектуальной системы онтология может использоваться для обеспечения:

- совместного использования разнородных данных и знаний;
- процесса решения задач, составляющих функциональность системы;
- лучшего понимания предметной области пользователями системы.

Лекция 2.3 - Онтологические модели представления знаний

РАСПРЕДЕЛЕННОЕ СОЗДАНИЕ ОНТОЛОГИЙ



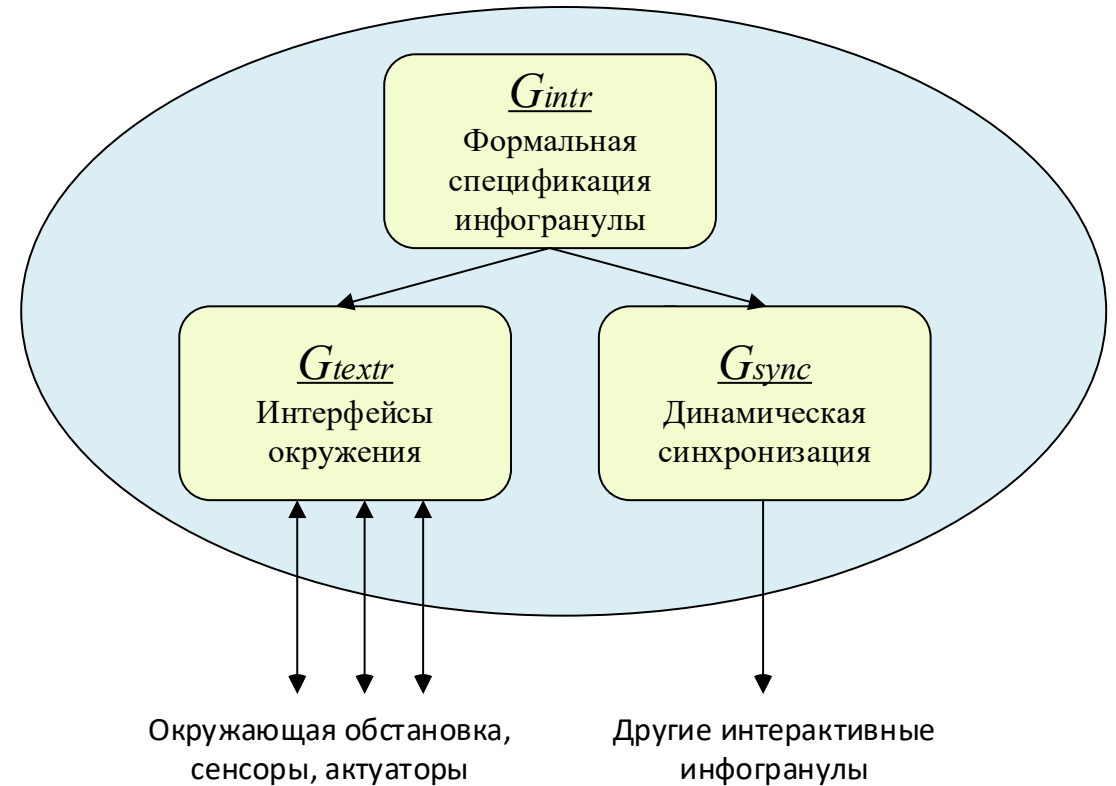
Лекция 2.3 - Онтологические модели представления знаний

ФОРМИРОВАНИЕ ОНТОЛОГИЙ НА ОСНОВЕ ИНТЕРАКТИВНОЙ ИНФОРМАЦИОННОЙ ГРАНУЛЯЦИИ

Суть таких гранул состоит в следующем.

Для совместного использования аппарата приближенных множеств и интерактивных вычислений гранула состоит из трех компонентов:

- интракомпонент, который отвечает за внутреннюю спецификацию инфогранулы;
- экстра-компонент, который определяет интерфейсы сенсорной связи инфогранулы с окружающим миром;
- синхрокомпонент, который отвечает за динамическую временную синхронизацию событий, происходящих с инфогранулой, то есть придает этому объекту свойства интерактивности.



Лекция 2.3 - Онтологические модели представления знаний

МЕРА КОНТЕКСТНО-ЗАВИСИМОЙ СЕМАНТИЧЕСКОЙ СХОЖЕСТИ ОНТОЛОГИЙ

Измерение сходства онтологий обычно понимается, определение степени похожести некоторых концептов из одной онтологии с некоторыми концептами из другой онтологии. Любая онтология, в отношении ее концептов может иметь структуру множества. Одной из самых известных мер подобия можно считать информационно-теоретическую меру, предложенную А. Тверским:

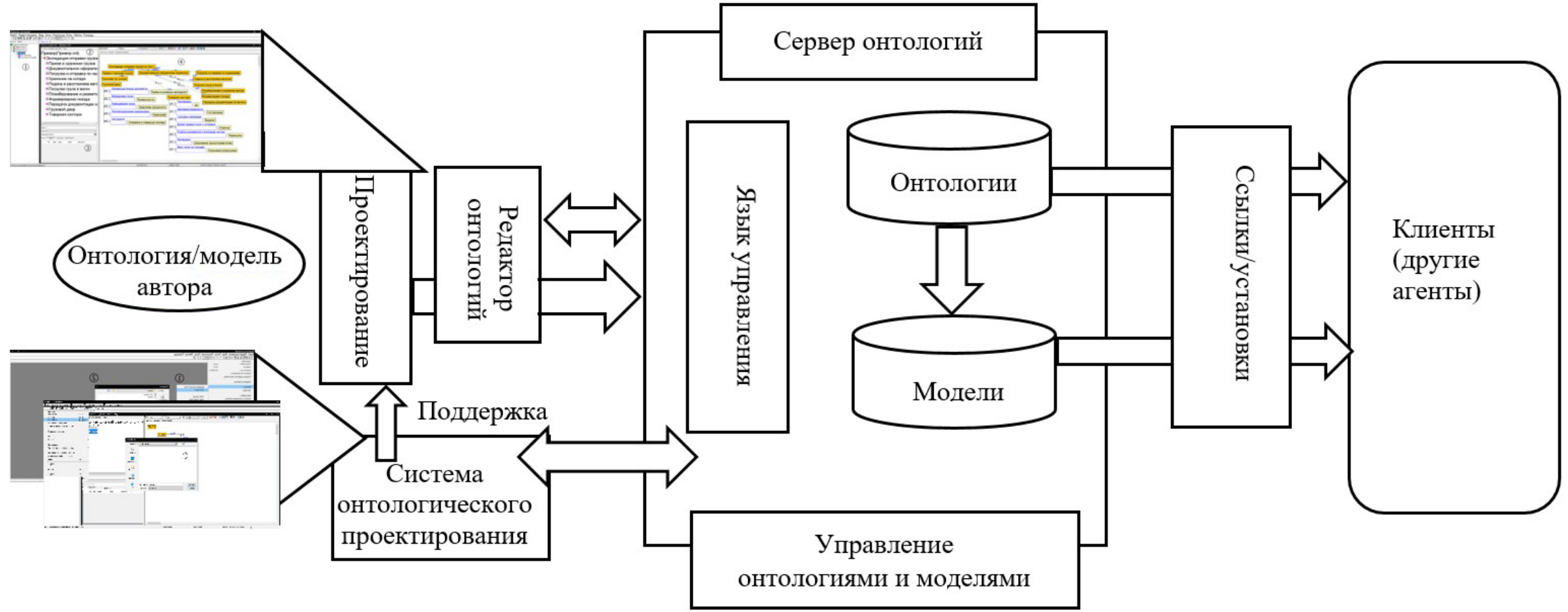
$$S_{ratio}(X, Y) = \frac{f(X \cap Y)}{f(X \cap Y) + \alpha f(X - Y) + \beta f(Y - X)}$$

Одной из первых известных информационно-теоретических мер, учитывающих контекстную схожесть онтологий является мера, предложенная П. Резником:

$$IC(c) = -\log p(c)$$

Лекция 2.3 - Онтологические модели представления знаний

АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РЕДАКТОРА ОНТОЛОГИЙ



Лекция 2.4 - Методы достоверного и правдоподобного вывода при принятии решений

ИНДУКТИВНЫЙ ВЫВОД

Индуктивный вывод - это вывод от частного к общему. Он позволяет на основании обобщения частных примеров выдвинуть гипотезу о существовании общей закономерности. В интеллектуальных системах, использующих индуктивный вывод, работает механизм, позволяющий при формировании гипотезы приписывать ей оценку правдоподобия (например, вероятность того, что данная гипотеза является истинной).

Часто приходится иметь дело с **массовыми событиями**. Необходимо установить устойчивое распределение основных признаков явления. Здесь возникает вопрос: до каких пределов можно уменьшить численность объектов с минимальным влиянием на результат обработки данных? Поэтому применяется выборка – один из видов неполной индукции. **Статистика** – это неполный вывод по индукции, в котором количественная информация о частоте признака в выборке переносится на все множество.

Тогда схема индуктивного вывода имеет примерно следующий вид:

Если $n\%$ выборки имеет данный признак, вероятно и $n\%$ всей совокупности будет иметь этот признак.

Лекция 2.4 - Методы достоверного и правдоподобного вывода при принятии решений

Вывод по аналогии

Вывод, основанный на перенесении рассуждения из исследованной области на другую область, похожую на исследованную. Если имеется вывод $A \rightarrow B$, и область, в которой определено A , гомоморфна области, где определена C , а область, где определено B , гомоморфна области, где определено D , то вывод $A \rightarrow B$ порождает вывод $C \rightarrow D$. Вывод по аналогии - это частный случай **правдоподобного вывода**.

Д. Поиа сформулировал два принципа вывода по аналогии:

- Предположение становится более правдоподобным, когда оказывается истинным аналогичное рассуждение;
- Предположение становится несколько более правдоподобным, когда становится более правдоподобным аналогичное предположение, а также схемы правдоподобных выводов:
- X аналогично Y . Если Y истинно, то Y более правдоподобно
- X аналогично Y . Если X более правдоподобно, то Y несколько более правдоподобно.

Лекция 2.4 - Методы достоверного и правдоподобного вывода при принятии решений

ВЕРОЯТНОСТНЫЙ ВЫВОД

Вероятностный вывод - это вывод, при котором каждое выражение, используемое в нем, имеет оценку правдоподобия в виде вероятности того, что оно является истинным.

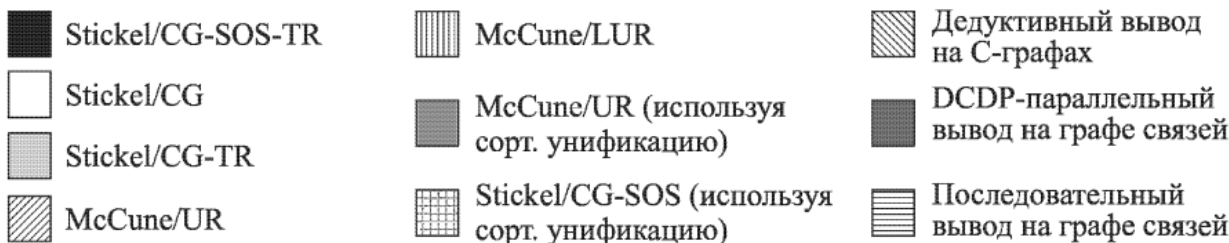
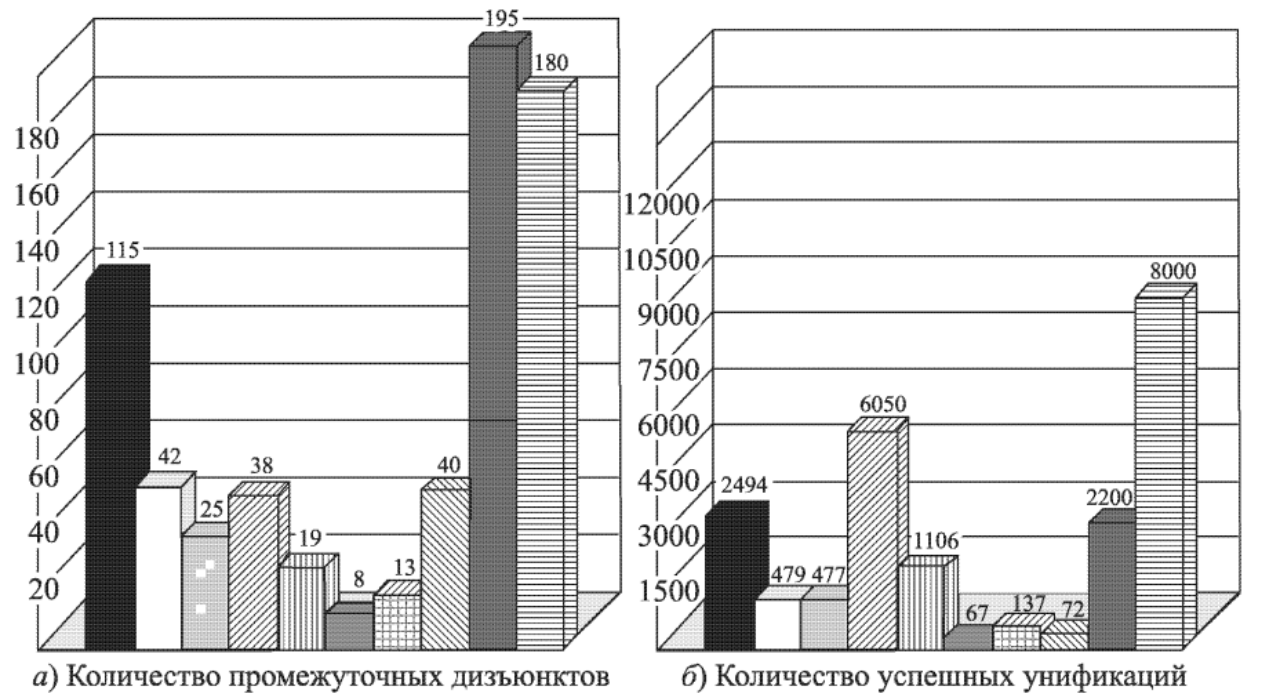
При вероятностном выводе применяются специальные процедуры для вычисления вероятности истинного значения результирующего выражения по вероятностям посылок, используемых при выводе, например байесовский вывод. Вывод базируется на теореме Байеса. Теорема Байеса подправляет вероятность гипотезы, данную новым свидетельством, следующим образом:

$$P(H|E) = \frac{P(E|H) P(H)}{P(E)}$$

где H представляет конкретную гипотезу, которая может быть, а может и не быть некоторой нулевой гипотезой. $P(H)$ называется априорной вероятностью H , которая была выведена прежде, чем новое свидетельство E стало доступным. $P(E|H)$ называется условной вероятностью наблюдения свидетельства E , если гипотеза H оказывается верной;

Лекция 2.4 - Методы достоверного и правдоподобного вывода при принятии решений

СРАВНЕНИЕ ЭФФЕКТИВНОСТИ ПРОЦЕДУР ДЕДУКТИВНОГО ВЫВОДА



CG — стратегия вывода с использованием графа, связей. SOS — вывод на графе связей с использованием целевого утверждения как множества поддержки. TR, — вывод с использованием метода Theory Links, который является расширением стандартного метода резолюции. UR, — вывод с использованием Unit-резолюции — модификации метода резолюции. LUR — вывод с использованием Linked Unit-резолюции — модификации метода UR-резолюции

Лекция 2.5 - Теория и системы аргументации

СИСТЕМЫ АРГУМЕНТАЦИИ И АБДУКТИВНЫЙ ВЫВОД

Абдукция – это процесс формирования объясняющей гипотезы. Точнее, при заданной теории и наблюдении, предложенном для объяснения, абдуктивный вывод должен определить одно или более наилучших объяснений наблюдения на основе заданной теории.

В искусственном интеллекте приложением-прототипом абдукции является задача диагностики.

Идея абдукции состоит в следующем:

Пусть имеется некоторая группа фактов и гипотез ***Th***, представляющих знания о предметной области. Если с их помощью можно объяснить имеющий место факт ***f***, то они принимаются.

Абдуктивный вывод имеет следующую схему:

Пусть ***h*** – объяснение факта ***f*** в ***Th***. Предположим, что ***Th&f*** – выполнимая формула, но ***f*** – не является следствием ***Th***, т.е. знания о предметной области ***Th*** не объясняют факта ***f***.

Необходимо вывести дополнительные факты ***h***, объясняющие ***f*** относительно ***Th***, т.е. ***Th, h*** \models ***f***, причем к фактам ***h*** предъявляются требования непротиворечивости и минимальности (абдуктивное объяснение ***h*** для ***f*** считается непротиворечивым, если ***Th&h*** – выполнимая формула; минимальным – если каждое объяснение, являющееся логическим следствием ***f***, эквивалентно ***h***).

Лекция 2.5 - Теория и системы аргументации

СИСТЕМЫ АРГУМЕНТАЦИИ И АБДУКТИВНЫЙ ВЫВОД

Примеры абдуктивного вывода:

- ❑ Всем известно, что трава во дворе дома будет мокрой после дождя или после работы дождевальная установки. Проснувшись утром, мы обнаружили, что трава мокрая. Соответственно, мы предлагаем гипотезу объяснения: ночью шел дождь либо работала дождевальная установка.
- ❑ Иоганн Кеплер пришел к заключению, что орбиты планет представляют собой эллипсы, на основании наблюдения того, что значения долготы орбиты Марса не соответствуют предполагаемой круговой орбите. Естественно, прежде чем предположить, что орбиты планет описываются эллиптически, он проанализировал несколько других гипотез, не прошедших последующих проверок. Известно, что (1) Все студенты из группы КТб04-2 юны, (2) Эти студенты юны. Следовательно, эти студенты являются студентами группы КТб04-2.
- ❑ Все люди смертны, Сократ смертен, следовательно, Сократ человек (может показаться, что здесь все нормально, но если вдуматься, то становится ясно, что вывод неправильный: из того, что Сократ смертен, вовсе не следует, что Сократ человек, ведь смертны и кошки, и собаки, и бабочки, и, может быть, деревья...).
- ❑ Все планеты круглые. Маша круглая. Значит, Маша - планета (полный бред, на самом деле, Маша - официант в пивном баре).

Вывод по абдукции успешно применяется для решения задач диагностики, понимания естественного языка, распознавания, накопления знаний, составления расписаний и, несомненно, является **74** очень важной составляющей интеллектуальных систем.

Лекция 2.5 - Теория и системы аргументации

СИСТЕМЫ ПЕРЕСМАТРИВАЕМОЙ АРГУМЕНТАЦИИ

Системы для отменяемой аргументации содержат (иногда неявно) следующие пять элементов:

- лежащий в основе логический язык;
- определение аргумента;
- определение конфликта между аргументами;
- определение поражения аргумента;
- определение оценки аргументов, которая может использоваться для определения понятия отменяемого логического следствия.

Аргументационные системы строятся над логическим языком и связанным с ним понятием отношения следования с помощью определения понятия аргумента. Это отношение следования предполагается монотонным.

Три базовые формы элементов:

- Иногда аргументы определяются как дерево выводов, основанное на посылках.
- Иногда – как последовательность таких выводов, то есть как дедукция.
- Некоторые системы определяют аргумент как пару: посылки – заключения, оставляя неявным то, как лежащая в основе логика осуществляет доказательство заключения по посылкам.

Лекция 2.5 - Теория и системы аргументации

СИСТЕМЫ ПЕРЕСМАТРИВАЕМОЙ АРГУМЕНТАЦИИ

Два типа **конфликтов** между аргументами:

- ❑ опровержение (*rebutting*) аргумента (у аргументов противоречивые заключения) – это симметричная форма атаки;
- ❑ подрыв (*undercutting*) аргумента – несимметричная форма атаки:
 - один аргумент делает предположение о недоказуемости (подобно тому, как это делается в логике умолчаний) и другой аргумент доказывает то, что предполагалось недоказанным первым аргументом;
 - один аргумент отрицает связь между посылками и заключением другого аргумента. Эта связь может быть нарушена только у недедуктивных аргументов, т.е. у индуктивных, абдуктивных или аргументов-аналогий.

Аргументы **по статусу** разбиваются на два класса:

- ❑ аргументы, с которыми можно выиграть спор (подтверждающие);
- ❑ аргументы, с которыми можно проиграть спор (отвергающие).

Лекция 2.5 - Теория и системы аргументации

ОСНОВНЫЕ СВОЙСТВА СЕМАНТИКИ, ОСНОВАННОЙ НА АРГУМЕНТАХ

Аргументы либо подтверждаются, либо не подтверждаются. Аргумент подтверждается, если все аргументы, поражающие его не подтверждаются. Аргумент не подтверждается, если он поражается подтвержденным аргументом.

Если аргументы равной силы интерферируют, то неясно, какой из аргументов останется непораженным.

НАЗНАЧЕНИЕ УНИКАЛЬНОГО СТАТУСА АРГУМЕНТАМ

Определения фиксированной точки. Поллок, Симари, Луи, Праккен и Сартор используют подход, связанный с понятием восстановления. Здесь учитывается следующее наблюдение: аргумент, пораженный другим аргументом, может быть подтвержден, если он восстанавливается третьим аргументом. В результате Данг определяет понятие приемлемости следующим образом:

- Аргумент A приемлем по отношению к множеству аргументов S тогда и только тогда, когда каждый аргумент, поражающий A , поражается аргументом из множества S .
- Пусть $Args$ – множество аргументов, упорядоченное бинарным отношением поражения. Тогда оператор F определяется так:

$$F: Pow(Args) \rightarrow Pow(Args) \text{ (Pow – степень множества);}$$
$$F(S) = \{A \in Args \mid A \text{ приемлем по отношению к } S\}.$$

Лекция 2.5 - Теория и системы аргументации

НАЗНАЧЕНИЕ УНИКАЛЬНОГО СТАТУСА АРГУМЕНТАМ

Продолжение...

- Аргумент подтверждается тогда и только тогда, когда он является элементом наименьшей фиксированной точки оператора F .
- Пусть $Args$ – множество аргументов, упорядоченное бинарным отношением поражения. Тогда оператор G определяется как:

$$G : Pow(Args) \rightarrow Pow(Args)$$

$$G(S) = \{A \in Args \mid A \text{ не поражается никаким аргументом в } S\}$$

- Определим уровни при доказательстве:
 - *все аргументы принадлежат множеству на уровне 0;*
 - *аргумент принадлежит множеству на уровне $n+1$ тогда и только тогда, когда он не поражается никаким аргументом на уровне n ;*
 - *аргумент подтвержден тогда и только тогда, когда существует такое значение m , что для любого $n \geq m$ аргумент принадлежит множеству уровня n .*

Лекция 2.5 - Теория и системы аргументации

НАЗНАЧЕНИЕ УНИКАЛЬНОГО СТАТУСА АРГУМЕНТАМ

Продолжение...

- ❑ Будем называть аргумент аннулированным тогда и только тогда, когда он не подтверждается, и либо самопоражается, либо поражается подтвержденным аргументом. Аргумент оправдываемый тогда и только тогда, когда он не подтверждается и не аннулируется.
- ❑ Поллок определяет подтверждение аргумента следующим образом:
 - ❑ аргумент находится на уровне 0 тогда и только тогда, когда он не поражает сам себя;
 - ❑ аргумент находится на уровне $n+1$ тогда и только тогда, когда он принадлежит уровню 0 и не поражается ни каким аргументом на уровне n ;
 - ❑ аргумент подтверждается тогда и только тогда, когда существует такое значение m , что для каждого $n \geq m$ аргумент принадлежит уровню n .
- ❑ Аргумент A подтверждается тогда и только тогда, когда выполняется:
 - ❑ A не поражает сам себя и
 - ❑ все подаргументы аргумента A подтверждаются и
 - ❑ все аргументы, поражающие аргумент A , поражают сами себя, или имеют по крайней мере один подаргумент, который аннулируется.

Лекция 2.5 - Теория и системы аргументации

ОБЗОР СИСТЕМ АРГУМЕНТАЦИИ

Аргументационной структурой (AF) является пара $(Args, defeat)$, где $Args$ – множество аргументов и $defeat$ – бинарное отношение на множестве $Args$. Структура AF называется конечной в том случае, когда каждый аргумент в $Args$ поражается, самое большее, конечным числом аргументов из $Args$. Множество аргументов является бесконфликтным, если ни один аргумент множества не поражается аргументами этого множества. Множество $Args$ может пониматься как множество всех аргументов, которые могут быть построены в данной логике из данного множества посылок.

В этой системе поражение интерпретируется в слабом смысле, т.е. как “атакует и не слабее”. Т.е. два аргумента могут поражать друг друга. Центральным понятием аргументационной структуры является **приемлемость**: “аргумент A приемлем по отношению к множеству аргументов S тогда и только тогда, когда каждый аргумент, поражающий A , поражается аргументом в S ”.

Еще одним важным понятием является **допустимость множества аргументов**: “Бесконфликтное множество аргументов S допустимо тогда и только тогда, когда каждый аргумент в S приемлем по отношению к S ”. Для нашего примера множества, $\{C\}$ и $\{A,C\}$ – допустимые, а все остальные подмножества для $\{A,B,C\}$ не будут допустимыми.

РАЗДЕЛ 3
ЭКСПЕРТНЫЕ СИСТЕМЫ
ПОДДЕРЖКИ ПРИНЯТИЯ
РЕШЕНИЙ

Лекция 3.1 – Введение в экспертные системы

Определение 1. **Экспертные системы** – это программные комплексы, аккумулирующие знания и опыт специалистов в некоторой предметной области с целью их тиражирования для консультации менее квалифицированных пользователей.

Определение 2. **Экспертные системы** – это класс программных систем, основанных на знаниях. Их вычислительные возможности определяются в первую очередь наращиваемой базой знаний и только во вторую очередь используемыми методами.

В связи с тем, что ЭС предназначены для решения неформализованных задач, для них характерны следующие особенности:

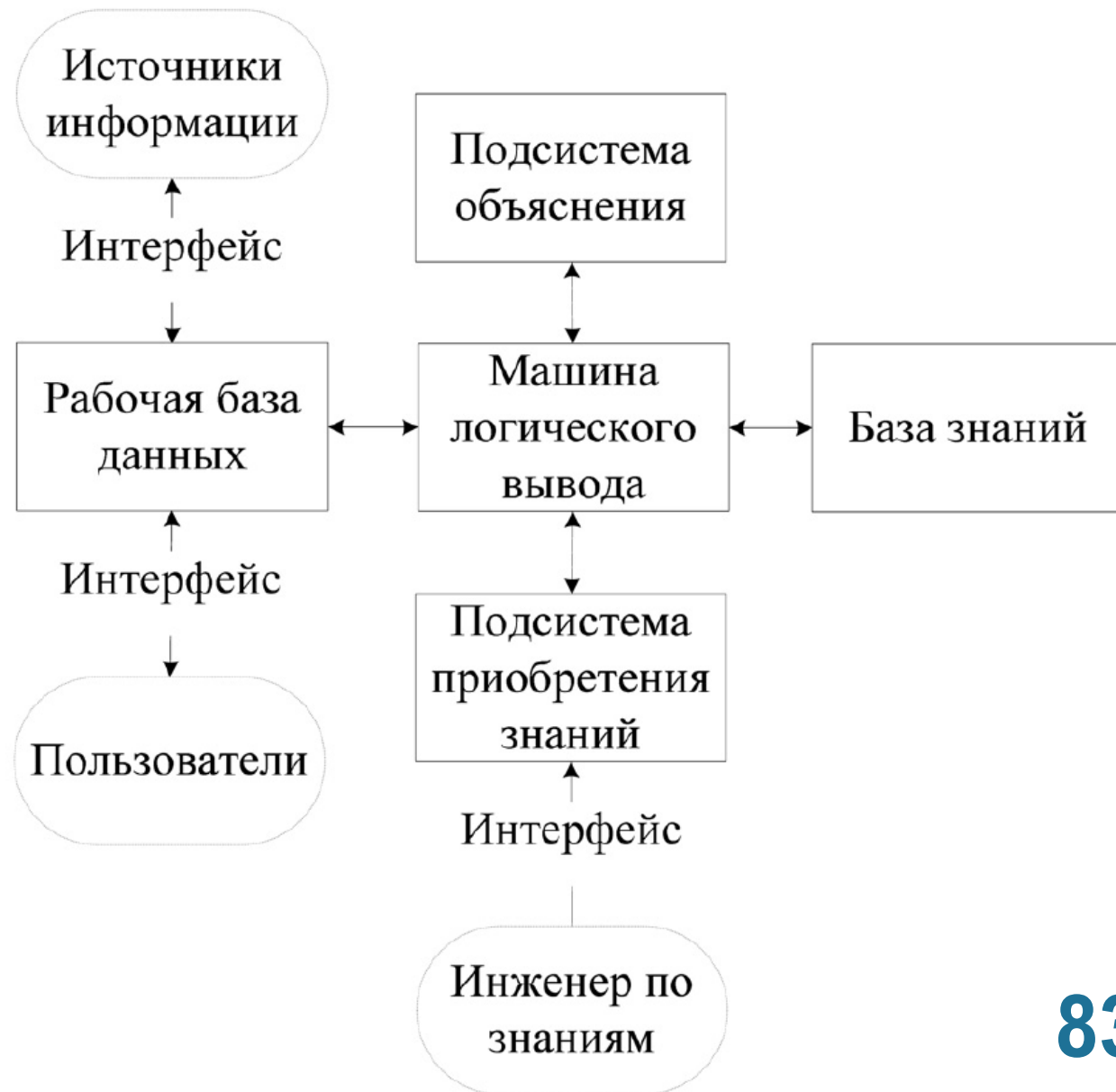
- Алгоритм решения задачи не известен заранее, а строится самой ЭС с помощью символических рассуждений, базирующихся на эвристических приемах;
- «Прозрачность» (ясность) полученных решений, т. е. система «осознает» в терминах пользователя, как она получила решение, и может проанализировать и объяснить свои действия и знания;
- Способность приобретения новых знаний от пользователя-эксперта и изменения в соответствии с ними своего поведения;
- Способность системы вести диалог о решаемой задаче на языке, понятном и удобном пользователю, т. е. ЭС должна обладать «дружественным» интерфейсом.

Лекция 3.1 - Введение в экспертные системы

АРХИТЕКТУРА ЭКСПЕРТНЫХ СИСТЕМ

В архитектуре ЭС выделяют информационную и программную части. В информационную часть входят: РБД (рабочая база данных) и БЗ (продукционная, сетевая, фреймовая или гибридная).

РБД – это множество фактов, описывающих текущее состояние предметной области. В программной части основными являются блоки: логического вывода (машина вывода), объяснения решений (выводов), приобретения знаний. Необходимы также интерфейсы между внешней средой и ЭС (интерфейс пользователя и инженера по знаниям с ЭС).



Лекция 3.1 - Введение в экспертные системы

ВИДЫ ЭКСПЕРТНЫХ СИСТЕМ

ЭС классифицируют по следующим основаниям:

- ❑ по типам задач: диагностика, мониторинг и оперативное управление, планирование, проектирование, прогнозирование, обучение, интерпретация и визуализация данных и др.;
- ❑ по типам инструментальных средств: объектно-ориентированные, Пролог-ориентированные, Лисп-ориентированные;
- ❑ по типам моделей представленных знаний: продукционные, фреймовые, сетевые, гибридные;
- ❑ по типам стратегии вывода: от ситуации – к цели (прямая стратегия вывода), от цели – к ситуации (обратная стратегия вывода), смешанные стратегии вывода;
- ❑ по типам машины вывода: дедуктивная, индуктивная, гибридная;
- ❑ по отношению ко времени: статические, когда не учитываются временные процессы, динамические, функционирующие в реальном масштабе времени, квазидинамические, когда временные процессы моделируются;
- ❑ по типу ЭВМ: суперкомпьютеры, большие ЭВМ, символьные процессоры, рабочие станции, ПЭВМ.

Лекция 3.2 - Построение баз знаний экспертных систем

Мощность ЭС определяется в первую очередь мощностью ее базы знаний и возможностью ее пополнения. Поэтому процесс построения базы знаний ЭС не только самый трудоемкий этап разработки ЭС, но и самый ответственный.

ПРИБРЕТЕНИЕ ЗНАНИЙ

Процесс получения знаний от эксперта (или другого источника знаний) и передача их экспертной системе называется **приобретением знаний**.

Источниками знаний для ЭС являются:

- человек-эксперт;
- тексты, в которых содержатся сведения о предметной области;
- эмпирические данные (таблицы, базы данных).

Важность процесса приобретения знаний обусловлена тем, что качество и эффективность решения задач ЭС определяется качеством и количеством используемых ею знаний.

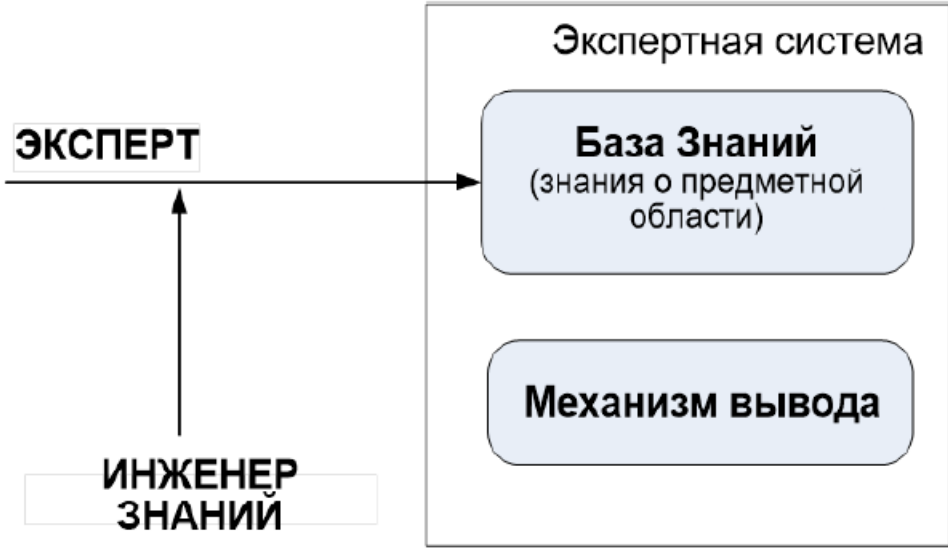
Сложность процесса приобретения знаний объясняется следующими факторами:

- объем знаний, используемых экспертом, достаточно велик и
- не все эти знания полностью осознаются экспертом.

Лекция 3.2 - Построение баз знаний экспертных систем

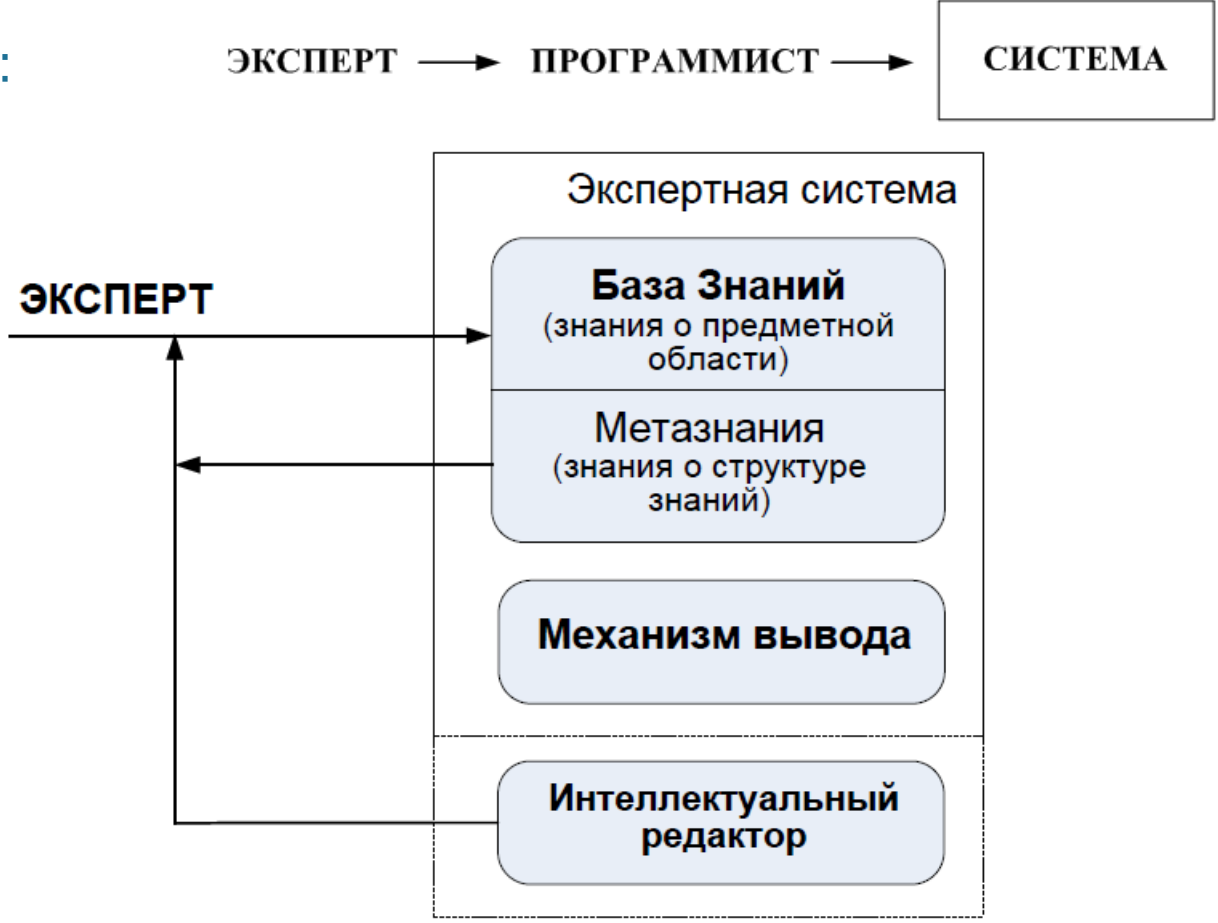
МОДЕЛИ ПРИОБРЕТЕНИЕ ЗНАНИЙ

Модель приобретения знаний ранними системами ИИ:



Модель приобретения знаний ЭС с помощью инженера знаний

ЭКСПЕРТ → ПРОГРАММИСТ → СИСТЕМА

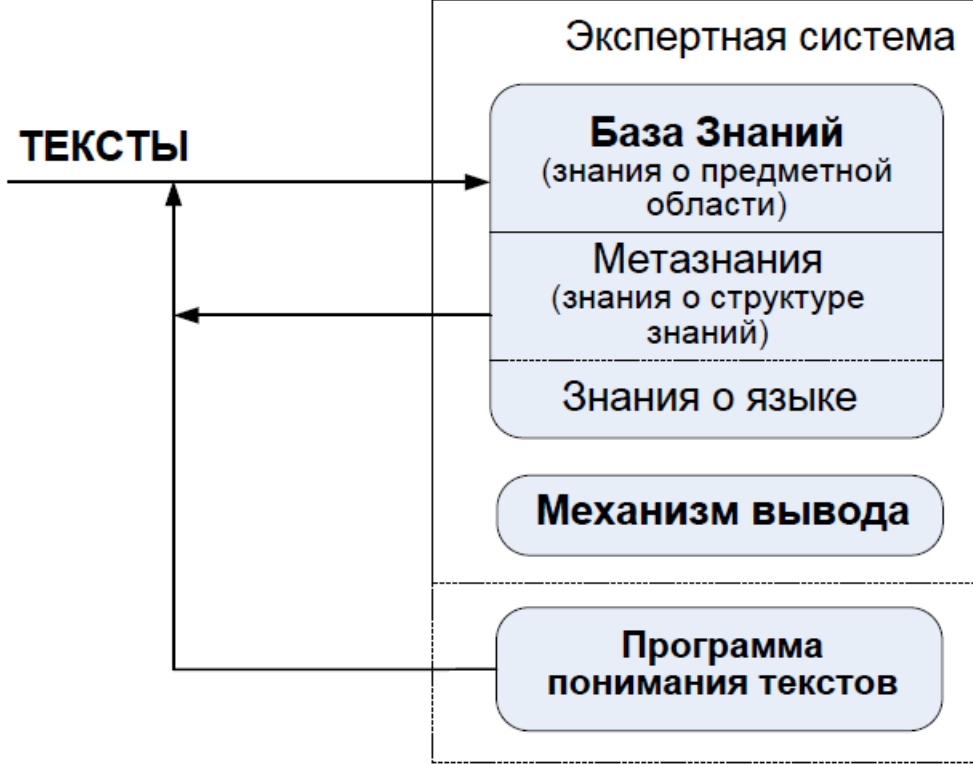
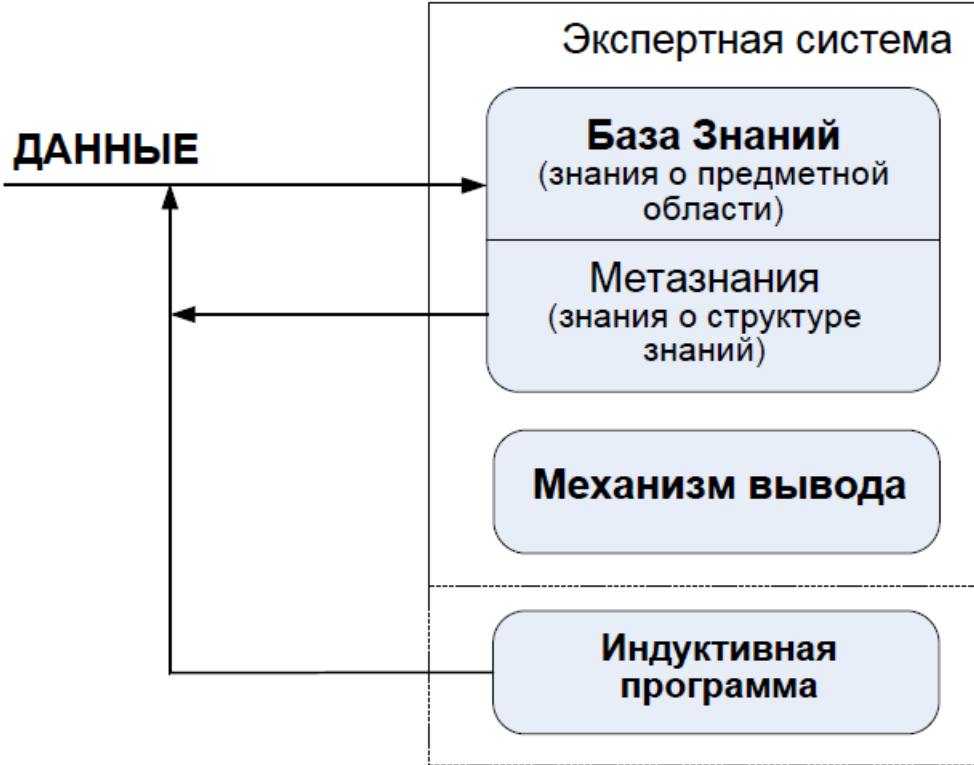


Модель приобретения знаний ЭС с помощью интеллектуального редактора

Лекция 3.2 - Построение баз знаний экспертных систем

МОДЕЛИ ПРИОБРЕТЕНИЕ ЗНАНИЙ

Модель приобретения знаний ЭС с помощью индуктивной программы:



Модель приобретения знаний ЭС с помощью программы понимания текста

Лекция 3.2 - Построение баз знаний экспертных систем

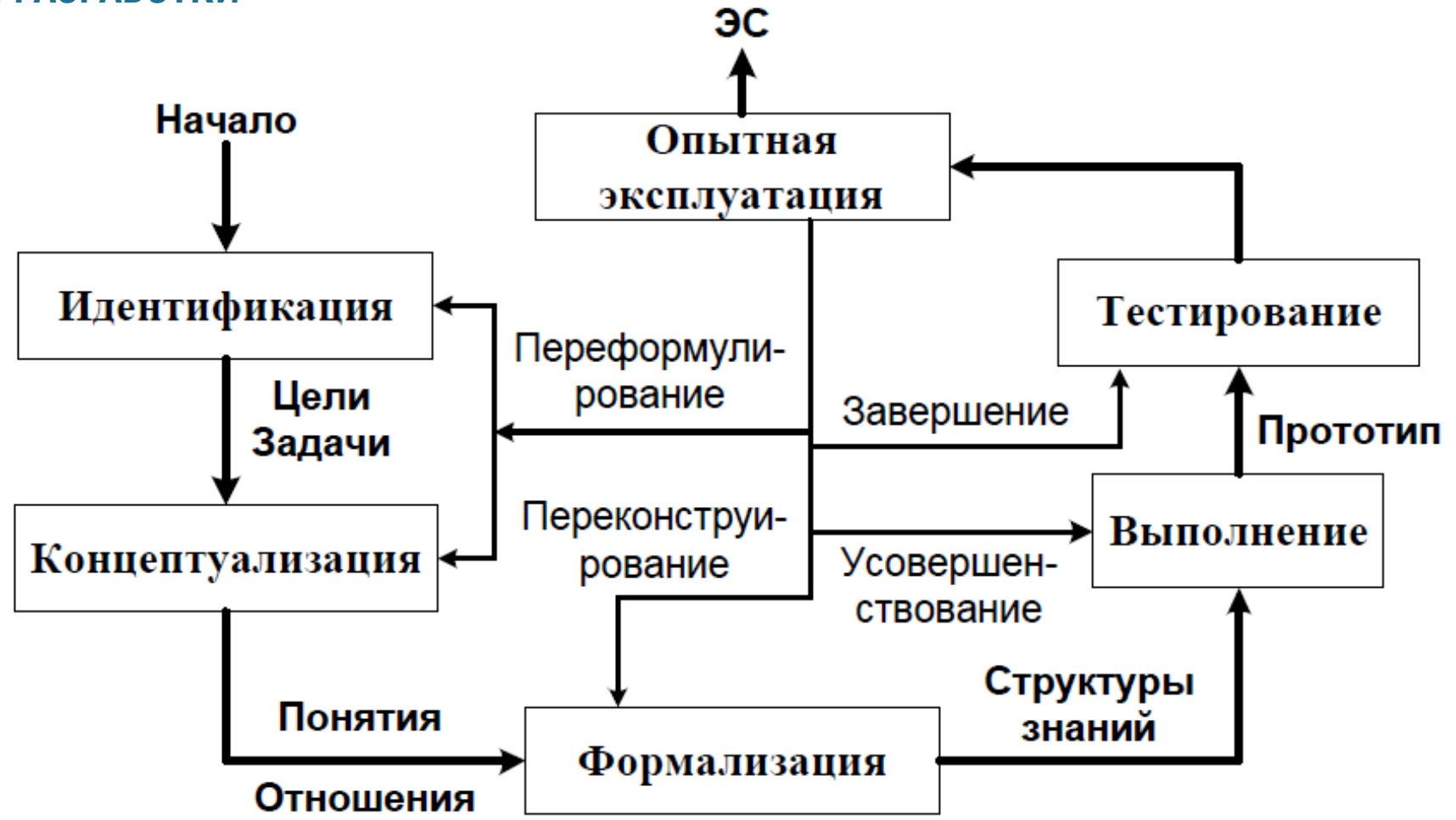
МЕТОДЫ ИЗВЛЕЧЕНИЯ ЗНАНИЙ

Классификация методов извлечения знаний:



Лекция 3.3 - Технология разработки экспертных систем

ОСНОВНЫЕ ЭТАПЫ РАЗРАБОТКИ



Разработка экспертной системы включает шесть этапов: *идентификацию, концептуализацию, формализацию, выполнение (реализацию), тестирование, опытную эксплуатацию.*

Лекция 3.3 - Технология разработки экспертных систем

ОСОБЕННОСТИ РАЗРАБОТКИ ЭКСПЕРТНЫХ СИСТЕМ

Чтобы разработка ЭС была **возможной** для данного приложения, необходимо одновременное выполнение, по крайней мере, следующих требований:

- существуют эксперты в данной области, которые решают задачу значительно лучше, чем начинающие специалисты;
- эксперты сходятся в оценке предлагаемого решения, иначе нельзя будет оценить качество разработанной ЭС;
- эксперты способны вербализовать и объяснить используемые ими методы, в противном случае трудно рассчитывать на то, что знания экспертов будут "извлечены" и вложены в ЭС;
- решение задачи требует только рассуждений, а не действий;
- задача не должна быть слишком трудной (т. е. ее решение должно занимать у эксперта несколько часов или дней, а не недель);
- задача хотя и не должна быть выражена в формальном виде, но все же должна относиться к достаточно «понятной» и структурированной области, т. е. должны быть выделены основные понятия, отношения и известны (хотя бы эксперту) способы получения решения задачи;
- решение задачи не должно в значительной степени использовать "здравый смысл" (т. е. широкий спектр общих сведений о мире и о способе его функционирования, которые знает и умеет использовать любой нормальный человек), так как подобные знания пока не удастся (в достаточном количестве) вложить в системы искусственного интеллекта.

Лекция 3.3 - Технология разработки экспертных систем

КЛАССИФИКАЦИЯ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ

Инструментальные средства разработки экспертных систем можно разделить на 4 категории:

- Языки программирования
 - традиционные языки (C, C++, C#, Java и др.),
 - языки символьной обработки (LISP, INTERLISP, SMALLTALK).
- Языки инженерии знаний (OPS-5, LOOPS, PROLOG).
- Программные обстановки, автоматизирующие разработку ЭС и интеллектуальных систем ИИ (KEE, ART, AGE, RLL, HEARSAY-III, TEIRESIAS, Semp-TAO).
- Оболочки экспертных систем – пустые ЭС, не содержащие ника-ких знаний о предметной области (EMYCIN, KAS, ЭКСПЕРТИЗА, ЭКО, ЭКСПЕРТ, DI*GEN).

При применении инструментария четвертого типа разработчик ЭС полностью освобождается от работ по программированию, так как он берет готовую базовую систему. Однако при использовании этого способа могут возникнуть следующие проблемы:

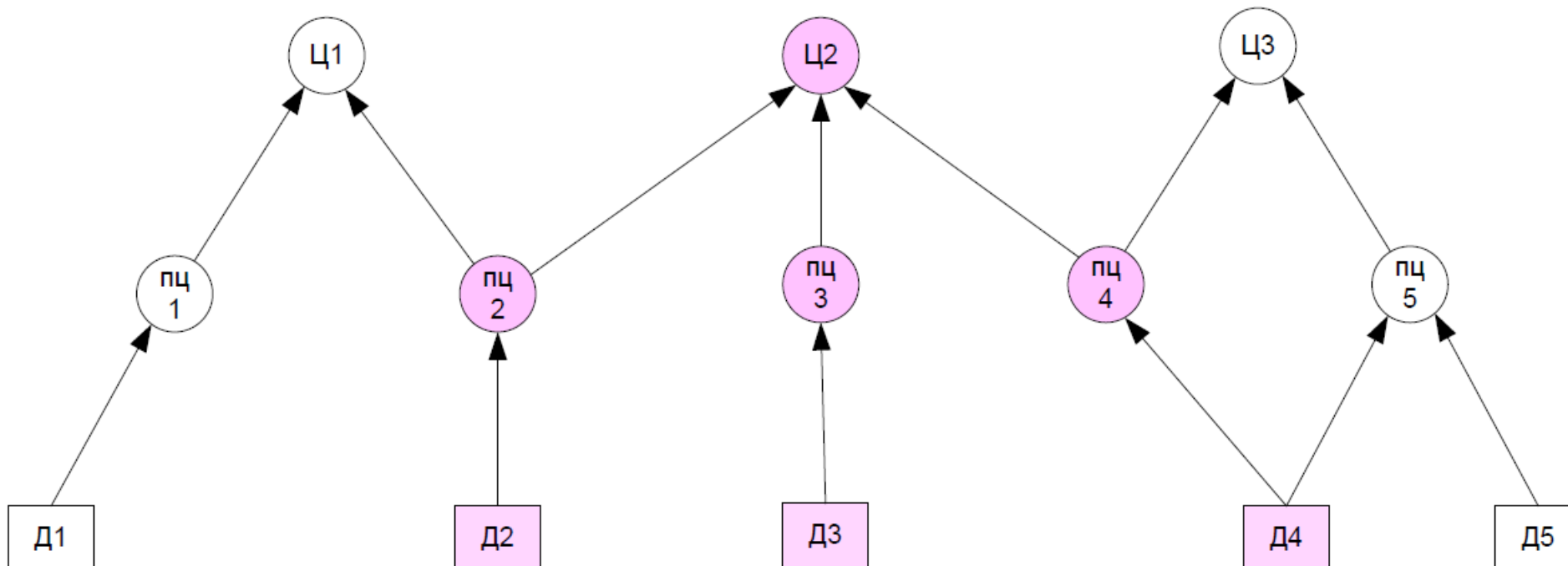
- управляющие стратегии, вложенные в процедуры вывода базовой системы, могут не соответствовать методам решения, которые использует эксперт, взаимодействующий с данной системой, что может приводить к неэффективным, а возможно, и неправильным решениям;
- язык представления знаний, принятый в базовой системе, может не подходить для данного приложения.

Лекция 3.4 - Объяснения в экспертных системах

Как правило, в ЭС поддерживается три вида объяснений:

- ❑ Объяснение действий (рассуждений) системы в ходе решения задачи;
- ❑ Ответы на вопросы о динамических знаниях системы;
- ❑ Ответы на вопросы о статических знаниях системы.

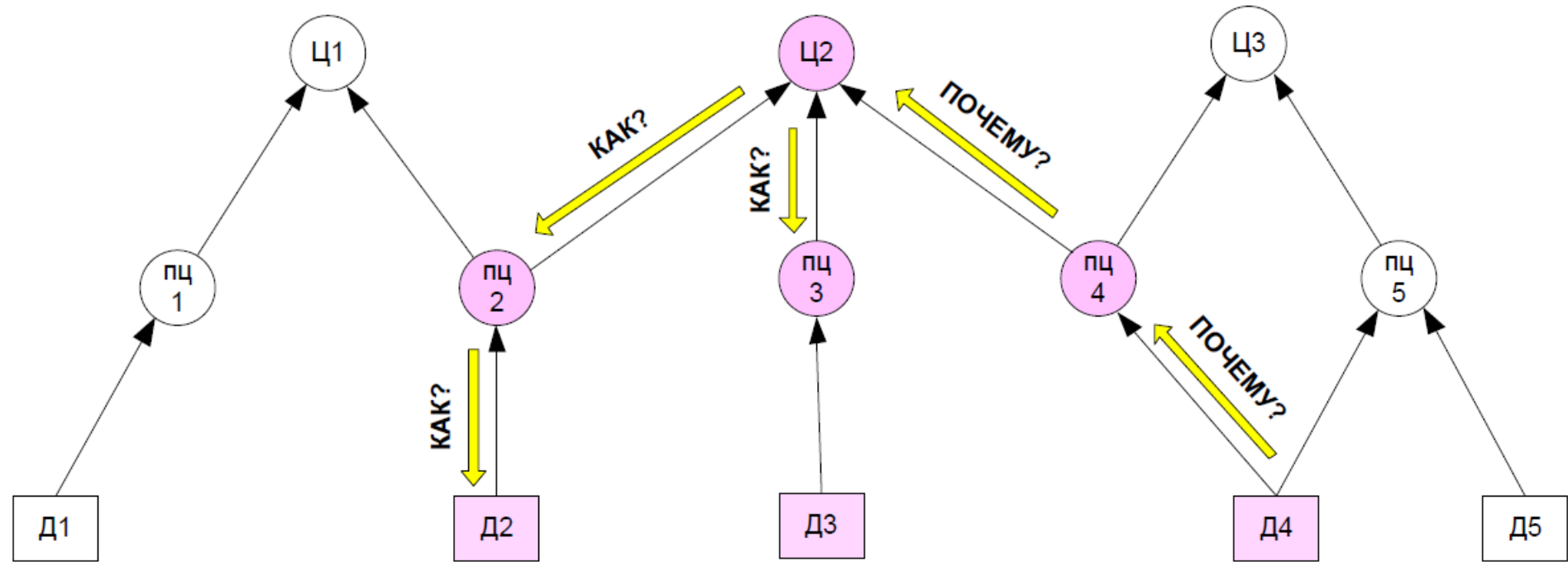
ПРИНЦИПЫ ПОСТРОЕНИЯ ПОДСИСТЕМЫ ОБЪЯСНЕНИЙ



Дерево целей

Лекция 3.4 - Объяснения в экспертных системах

ПРИНЦИПЫ ПОСТРОЕНИЯ ПОДСИСТЕМЫ ОБЪЯСНЕНИЙ



Навигация по дереву целей

Ответы на вопросы о динамических знаниях системы сводятся к поиску соответствующей информации в рабочей памяти ЭС.

Ответы на вопросы о статических знаниях ЭС требуют поиска всех правил, посылки и действия которых соответствуют обрабатываемому вопросу.

Лекция 3.4 - Объяснения в экспертных системах

ОСНОВНЫЕ ДОСТОИНСТВА И НЕДОСТАТКИ ПОДСИСТЕМЫ ОБЪЯСНЕНИЙ

Основные **достоинства подсистемы объяснений** в ЭС:

- объяснения помогают пользователю использовать систему для решения своих задач;
- объяснения повышают степень доверия пользователя к ЭС, позволяя пользователю убедиться в правильности полученных результатов;
- объяснения служат для обучения пользователя;
- объяснения служат для отладки базы знаний ЭС.

Основные **недостатки подсистемы объяснений** в ЭС:

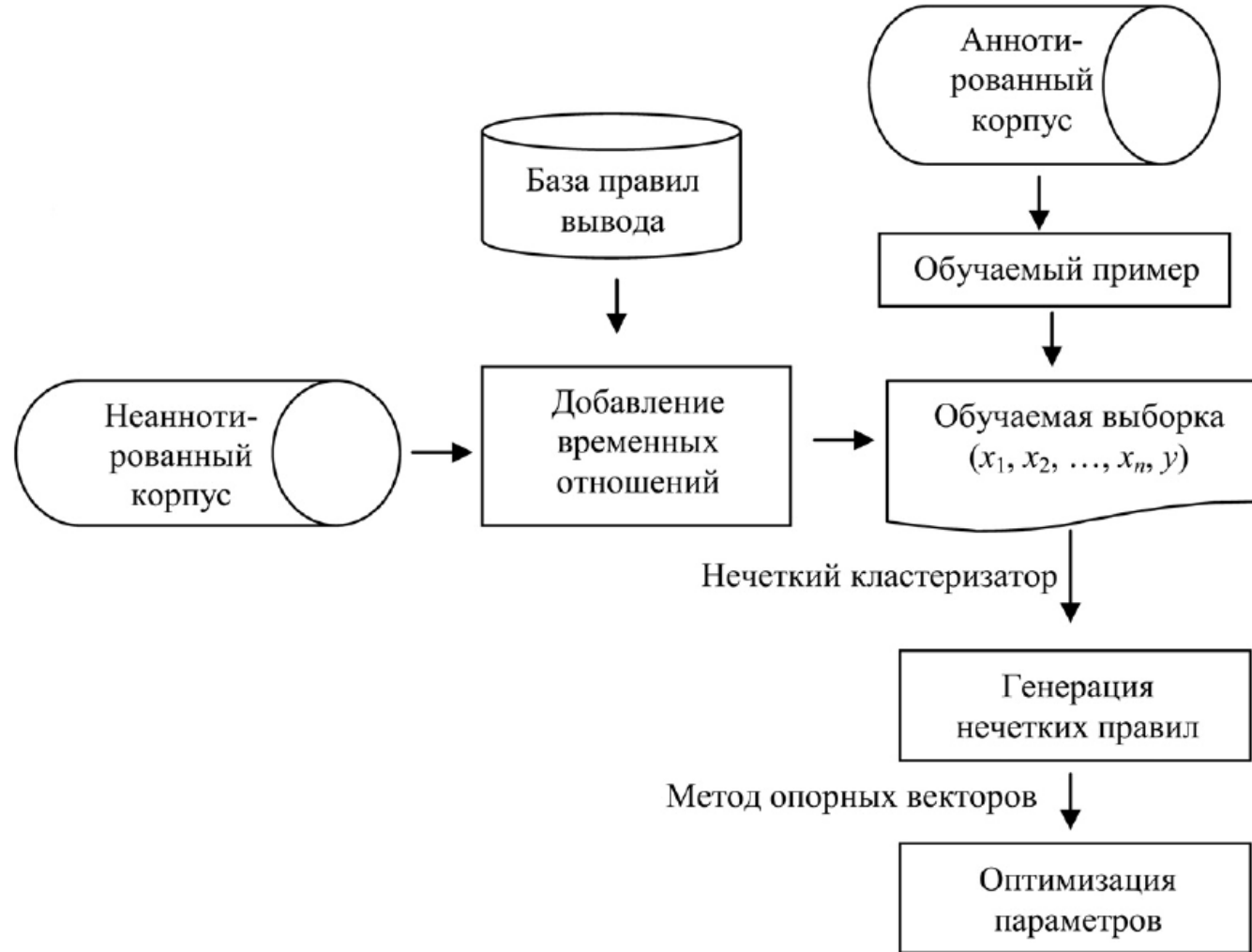
- запросы на объяснение интерпретируются только в одном узком смысле (вопросы ПОЧЕМУ и КАК интерпретируются только в терминах целей и правил);
- не все действия системы могут быть объяснены (например, почему сначала проверялась одна гипотеза, а потом другая);
- объяснения основываются, фактически, на треке выполнения программы, поэтому при смене интерпретатора необходимо менять и систему объяснений.

Лекция 3.5 – Алгоритмы принятия решений в экспертных системах



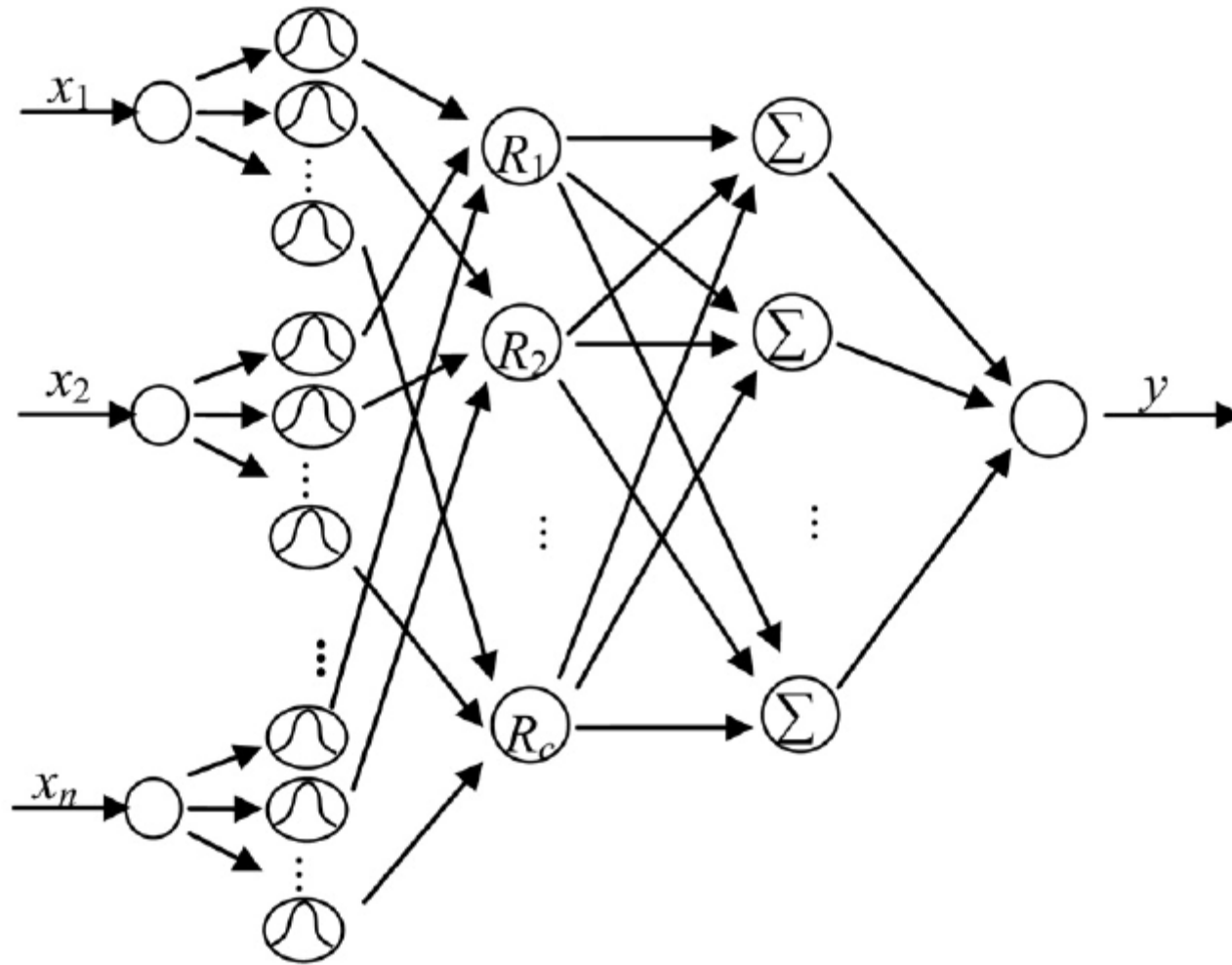
Алгоритм нечёткой экспертной системы

Лекция 3.5 – Алгоритмы принятия решений в экспертных системах



Алгоритм системы обработки временной информации

Лекция 3.5 – Алгоритмы принятия решений в экспертных системах



Алгоритм нечеткой нейронной сети

Лекция 3.6 – Разработка НМИ экспертных систем

The screenshot displays the Protégé 3.4 rc1 interface for editing an ontology. The main window title is "newspaper Protégé 3.4 rc1 (file:/work/protége/Protege_3.4_rc1/examples/newspaper/newspaper.pprj)". The menu bar includes File, Edit, Project, Code, Window, Collaboration, Tools, and Help. The toolbar contains icons for file operations and editing. The interface is divided into several panes:

- CLASS BROWSER:** Shows the project "newspaper" and a hierarchy of classes: :THING, :SYSTEM-CLASS, Author, Content, Layout_info, Library, Newspaper (selected), Organization, and Person.
- CLASS EDITOR:** Shows the selected class "Newspaper" with the following details:
 - Name: Newspaper
 - Role: Concrete
 - Template Slots table:

Name	Cardinal...	Insta
contents	multiple	Insta
date	single	Strin
number_of_pages	single	Integ
prototype	single	Insta
- Collaboration:** Shows a "Notes for Newspaper" section with a filter set to "By author...", a "Go" button, and options for "New Thread" or "Reply using a Comment".

Лекция 3.6 – Разработка НМІ экспертных систем

The screenshot displays the OWL Viewer interface for the 'core' ontology. The main window shows a graph of classes and properties. Key elements include:

- Classes:** Thing (dashed circle), Automatic Tag (red circle), UserAccount (blue circle), Subclass (blue circle), Tag (blue circle), Tagging (blue circle), Private Tagging (blue circle), Literal (yellow rectangle), and Concept (blue circle).
- Properties:** has creator, about, created, creator, modified, tag label (functional), next tag (functional), automatic tag me, meaning of (functional), tag created (functional), has creator (functional), creator of, taggran (functional), tagged with, tag, has access, tagging created (functional), tagging modified, and dateTime.
- Relationships:** Automatic Tag is a subclass of Tag. UserAccount is a subclass of Tag. Tagging is a subclass of Tag. Private Tagging is a subclass of Tagging. Literal is a subclass of Thing. Concept is a subclass of Tag.

The interface includes a menu bar (File, Edit, View, Reasoner, Tools, Refactor, Window, Help), a search bar for entities, and a sidebar with 'VOWL Controls' (Class Distance, Datatype Distance, Pause Layouting, Reset). The version is Beta 0.1, and the URL is <http://vowl.visualdataweb.org>. A footer note says: 'To use the reasoner click Reasoner->Start reasoner' with a checked 'Show Inferences' option.

Лекция 3.6 – Разработка НМИ экспертных систем

The screenshot displays the VOWL Viewer interface for the ontology `core` (URI: `http://purl.org/muto/core`). The main window shows a graph with the following elements:

- A light blue circle labeled "Private Tagging".
- A dark blue circle labeled "UserAccount... (external)".
- A red-bordered box containing the text "has creator (functional) creator of".
- A dotted arrow labeled "Subclass" points from "Private Tagging" to a partially visible light blue circle at the top.
- A red arrow points from the "has creator" box to the "UserAccount..." circle.
- A yellow circle is present on the right side of the graph.

The VOWL Sidebar on the right provides details for the selected property:

- Type:** Object Property
- Name:** has creator
- URI:** `http://purl.org/muto/core#hasCreator`
- Comment:** Every tagging is linked to at most one user account. This property can be omitted for automatic taggings. In contrast to its superproperty `sioc:has_creator`, it is functional and with an explicit domain. Use `sioc:member_of` to define groups for group tagging or link to `foaf:Agent`, `foaf:Person`, or `foaf:Group` via `sioc:account_of`.
- Domain:** Tagging
- URI:** `http://purl.org/muto/core#Tagging`

The VOWL Controls panel at the bottom right includes sliders for "Class Distance" and "Datatype Distance", and buttons for "Pause Layouting" and "Reset". The version is Beta 0.1, with a link to `http://vowl.visualdataweb.org`.

At the bottom of the window, there is a checkbox for "Show Inferences" which is checked, and a note: "To use the reasoner click Reasoner->Start reasoner".

Лекция 3.6 – Разработка НМІ экспертных систем

The screenshot displays the VOWL Viewer interface for the ontology `core` (URI: `http://purl.org/muto/core`). The main window shows a class hierarchy where `Tagging` is a subclass of `Item` (external). `Tagging` is associated with several properties: `tagging created (functional)`, `tagging modified`, and `has tag`. `Tagging` is also linked to `Thing` via `tagged resource (functional)` and `tagged with`. A `dateTime` property is also shown. The VOWL Sidebar on the right provides version information (Version 1.0) and domain details (Domain: `http://purl.org/muto/core#Tagging`). The VOWL Controls panel includes sliders for Class Distance and Datatype Distance, and buttons for Pause Layouting and Reset.

core (http://purl.org/muto/core) : [http://muto.socialtagging.org/core/v1.rdf]

File Edit View Reasoner Tools Refactor Window Help

core (http://purl.org/muto/core) Search for entity

Active Ontology Entities Classes Object Properties Data Properties Individuals OWLviz DL Query OntoGraf SPARQL Query VOWL

VOWL Viewer: VOWL Sidebar: VOWL Controls:

Version 1.0: The range information has been removed for greater flexibility and OWL Lite conformance (no longer owl:unionOf). Classes from different vocabularies can now be used here - such as `sic:UserAccount`, `sic:Usergroup`, `foaf:OnlineAccount`, `foaf:Group`, or `dcterms:Agent` -, though we recommend the use of `sic:UserAccount` or `sic:Usergroup` to remain in the SIOC namespace.

Domain: Tagging <http://purl.org/muto/core#Tagging>

has access

Class Distance: Datatype Distance: Pause Layouting Reset

Version: Beta 0.1 <http://vowl.visualdataweb.org>

To use the reasoner click Reasoner->Start reasoner Show Inferences

Лекция 3.6 – Разработка НМІ экспертных систем

The screenshot displays the VOWL Viewer interface for the 'core' ontology. The main window shows a hierarchical ontology diagram with classes like 'Tagging', 'Tag', 'Item', 'Automatic Tag', 'Private Tagging', 'User Account', and 'Concept'. Properties such as 'tagging created', 'tagging modified', 'tag of', and 'has tag' are visible. A yellow circle highlights a specific area in the diagram. The right sidebar shows the 'VOWL Sidebar' with the following details:

- Type: OWLThing
- Name: Thing
- URI: <http://www.w3.org/2002/07/owl#Thing>

The 'VOWL Controls' section includes sliders for 'Class Distance' and 'Datatype Distance', and buttons for 'Resume Layouting' and 'Reset'. The version is Beta 0.1, and the URL is <http://vowl.visualdataweb.org>. At the bottom, there are instructions: 'To use the reasoner click Reasoner->Start reasoner' and a checked checkbox for 'Show Inferences'.

Лекция 3.6 – Разработка НМИ экспертных систем

The screenshot displays the VOWL Viewer application interface. The main window shows a complex ontology graph with various classes and properties. The graph is organized into several clusters, with a central cluster of classes like 'Online Account', 'Online Gaming...', and 'Online Chat...'. Other clusters include 'Project', 'Group', and 'Organis...'. The graph is rendered in a circular layout, with nodes connected by lines representing relationships. The interface includes a menu bar (File, Edit, View, Reasoner, Tools, Refactor, Window, Help), a search bar for entities, and a sidebar with controls for Class Distance and Datatype Distance. The VOWL Controls section has a yellow highlight around the Class Distance slider. The version information is Beta 0.1, and the URL is <http://vowl.visualdataweb.org>.

Лекция 3.6 – Разработка НМІ экспертных систем

The screenshot displays the VOWL Viewer interface for an ontology. The main window shows a network of classes and their relationships:

- Document** (CreativeWork): A central class with properties like `primary topic`, `workplace homepage`, `publications`, `work info homepage`, `sha1sum (hex)`, `tipjar`, `openid (inverseFunctional)`, `interest`, `weblog (inverseFunctional)`, `account service homepage`, and `account`. It is a subclass of `Online Account`.
- Agent** (Agent): A class with properties like `gender (functional)`, `Skype ID`, `age (functional)`, `status`, `topic_interest`, `personal mailbox (inverseFunctional)`, `Yahoo chat ID`, `AIM chat ID`, `personal mailboxes`, `QQ chat ID`, and `birthday (functional)`. It is a subclass of `Document`.
- Online Account**: A class with properties like `account service homepage` and `account`.
- Literal**: Several instances of literals are shown, connected to the `Document` and `Agent` classes via various properties.

The VOWL Sidebar on the right provides details for the selected `Document` class:

Type:	Equivalent Class
Name:	Document
	http://xmlns.com/foaf/0.1/Document
Comment:	A document.
Equivalent Class:	CreativeWork
	http://schema.org/CreativeWork

The VOWL Controls panel at the bottom right includes sliders for `Class Distance` and `Datatype Distance`, and buttons for `Pause Layouting` and `Reset`. The version is Beta 0.1, and the URL is <http://vowl.visualdataweb.org>.

Лекция 3.7 – Инструментальные средства экспертных систем

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА УПРАВЛЕНИЯ ОНТОЛОГИЯМИ

Критерии	Средства				
	<i>Ontolingua</i>	<i>Protégé</i>	<i>OilEd</i>	<i>OntoEdit</i>	<i>Web-DESO</i>
Разработчики, проекты	Лаборатория систем знаний Университета Стэнфорда	Проект « <i>Semantic Web</i> », лаборатория медицинской информатики Университета Стэнфорда	Проект « <i>On-To-Knowledge-Project</i> » Университета Манчестера	Проект « <i>Semantic Web</i> », компания <i>Ontoprise GmbH</i>	«Система интеграции знаний», Санкт-Петербургский институт информатики и автоматизации РАН
Формат / формализм представления знаний	Логика первого порядка	<i>OKBC</i> – совместимая фреймовая модель знаний	<i>Description Logic</i>	<i>RDF</i> – совместимая фреймовая модель знаний	Объектно-ориентированные сети ограничений
Методы моделирования понятий и отношений	Набор аксиоматизированных таксономий и отношений между ними	Сложная таксономия	Сложная таксономия, иерархия	Сложная таксономия, иерархия	Таксономия, иерархия, ассоциативные отношения

Лекция 3.7 – Инструментальные средства экспертных систем

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА УПРАВЛЕНИЯ ОНТОЛОГИЯМИ

Методы описания области допустимости применения знаний	Аксиомы, теории	Аксиомы, ограничения	База правил	Аксиомы как сущности первого порядка	Совместимость классов, функциональные отношения
Поддерживаемые форматы	<i>Loom, Epikit, Algernon, KIF, CycL, EXPRESS</i>	<i>RDF</i>	<i>RDF/RDF-схема, DAML+OIL</i>	<i>DAML+OIL, FLOGIC, SQL2</i>	<i>DAML+OIL</i>
Возможности многократного использования онтологий	<i>Включение всего содержимого внешней онтологии без изменений; ограничение числа включаемых аксиом; полиморфное уточнение расширения включаемых аксиом посредством уточнения их операторов</i>	<i>Включение, соединение (соединение, учитывая разрешение конфликтов между именами понятий и структурным представлением)</i>	<i>Включение</i>	–	<i>Соединение</i>

Лекция 3.7 – Инструментальные средства экспертных систем

Универсальные языки программирования (*Java*).

Языки, «ориентированные на знания»:

языки представления знаний (*KIF*);

языки переговоров и обмена знаниями (*KQML*, *AgentSpeak*, *April*);

языки спецификации агентов.

Специализированные языки программирования агентов (*TeleScript*).

Языки сценариев и *scripting languages* (*Tcl/Tk*).

Символьные языки и языки логического программирования (*Oz*).