

Представление знаний в ИИ. Базы знаний.

Молодченков Алексей Игоревич

Этапы создания прикладных систем

- *Идентификация проблемы.*
- *Концептуализация проблемы.*
- *Формализация проблемы.*
- *Реализация.*
- *Тестирование.*

Системы, основанные на правилах или продукционные системы

Основные компоненты:

- *рабочая память* (или, как иногда говорят, *глобальную базу данных*),
- *множество правил*, выполняющих некоторые действия (во внешней среде и в рабочей памяти)
- *стратегия управления*, в соответствии с которой происходит выбор правил для применения и выполнение действий

С точки зрения архитектуры такой подход обладает следующими **существенными отличиями** от архитектур традиционных программных систем:

- рабочая память доступна всем правилам;
- отсутствуют вызовы правил из других правил;
- отсутствует априорно заданный алгоритм решения задачи (т.е. порядок выполнения правил) – алгоритм решения задачи является одним из результатов её решения;
- данные и результаты вычислений становятся доступными правилам только через рабочую память.

Правила для представления знаний

Правилом включает в себя:

- *Имя правила;*
- *Условия;*
- *Тело правила.*

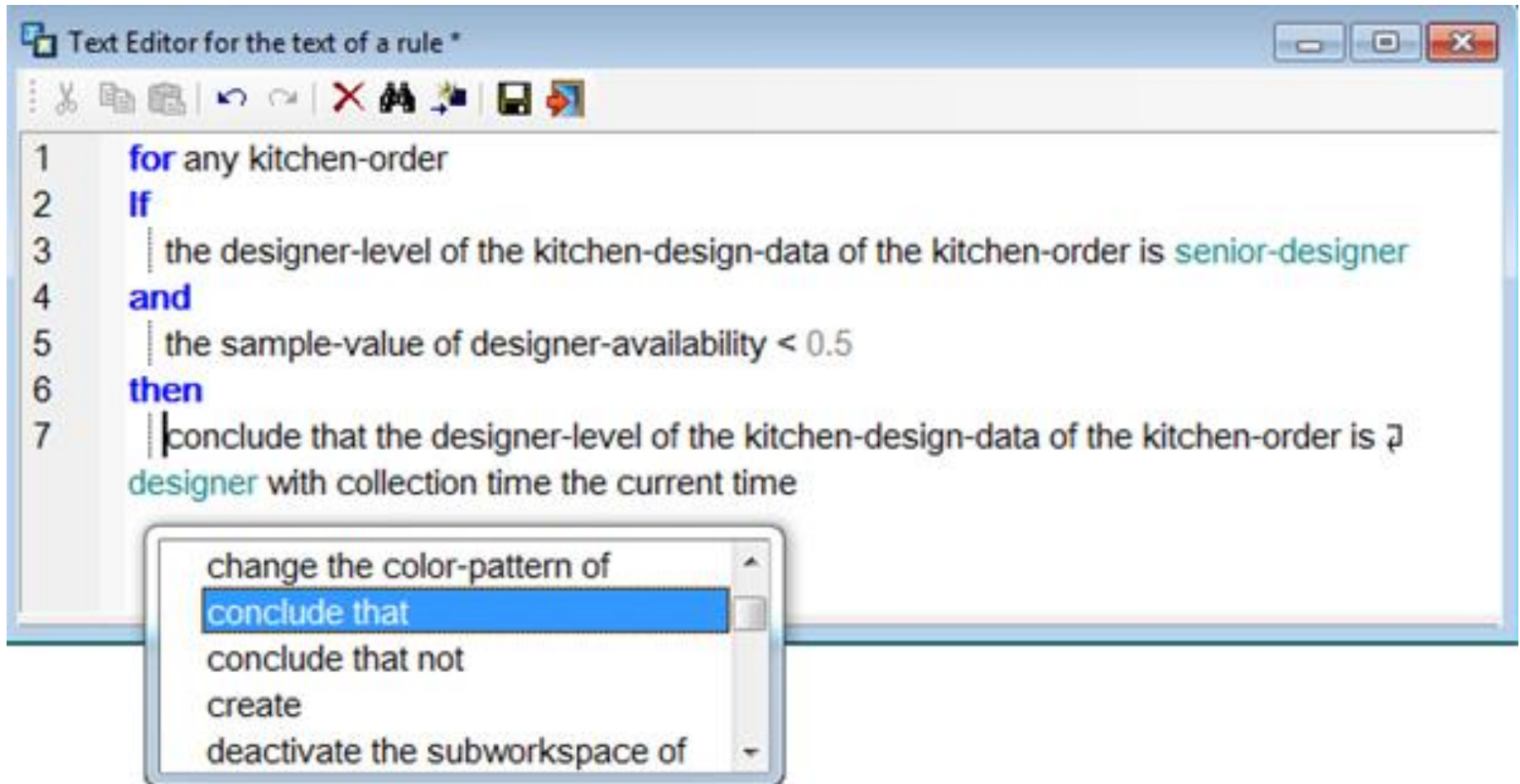
Стратегии управления

- Шаг 1. Выбрать очередное правило из множества правил;
- Шаг 2. Проверить выполнимость условия правила в текущем состоянии рабочей памяти;
- Шаг 3. Если условие правила выполнено, поместить правило в конфликтное множество;
- Шаг 4. Если множество применимых правил исчерпано, выбрать какое-либо правило из конфликтного множества правил и применить его.
- Шаг 5. Перейти к шагу 1.

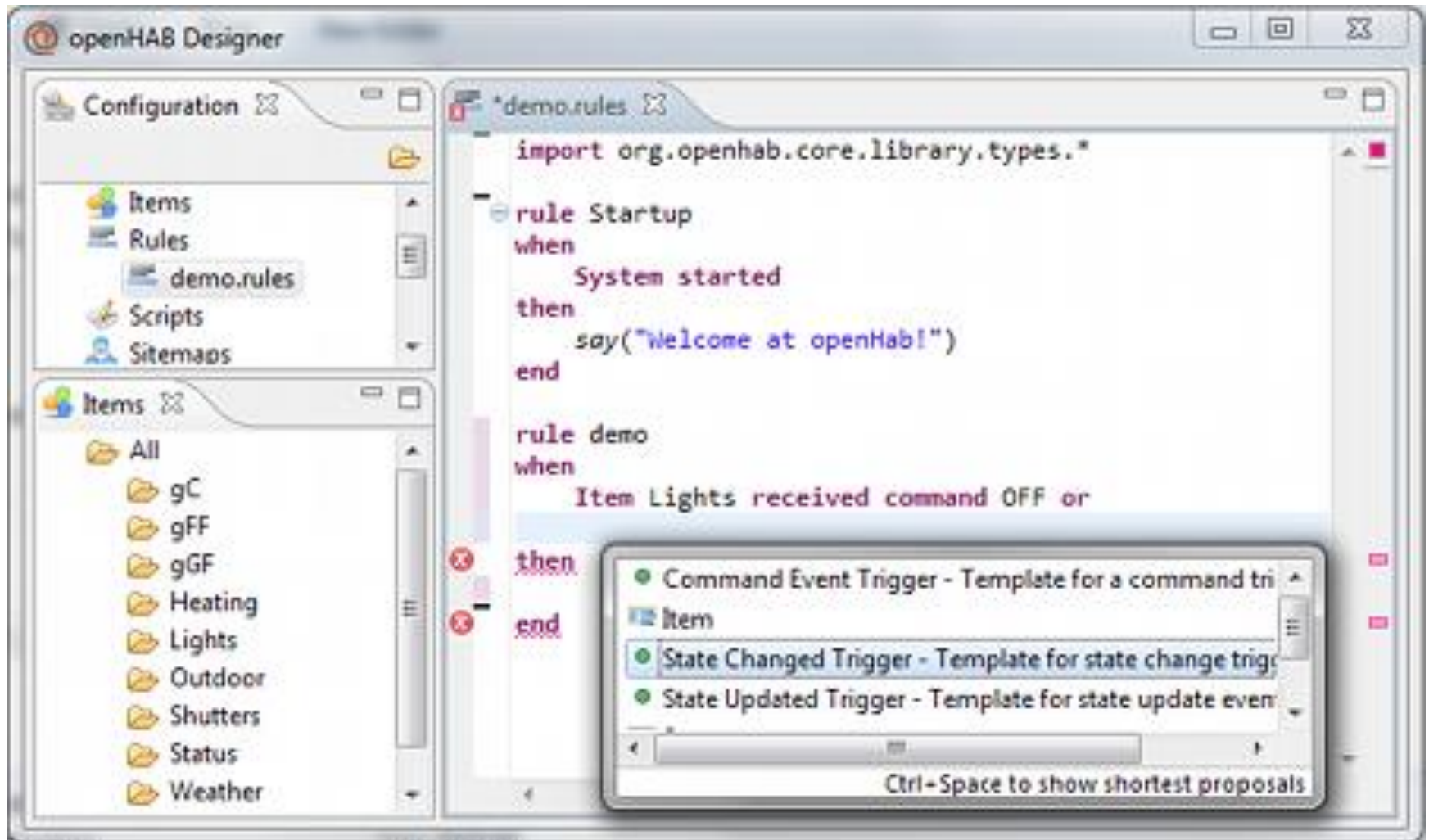
В дальнейшем тройку: множество правил + рабочая память + стратегия управления будем называть системой, основанной на правилах.

Состоянием системы, основанной на правилах, будем называть состояние рабочей памяти вместе с множеством применимых правил.

G2 редактор правил



openHAB редактор правил



Drools, пример правила

```
rule "Tests for type1 machine"  
  salience 100  
  when  
    machine : Machine( type == "Type1" )  
  then  
    Test test1 = testDAO.findByKey(Test.TEST1);  
    Test test2 = testDAO.findByKey(Test.TEST2);  
    Test test5 = testDAO.findByKey(Test.TEST5);  
    machine.getTests().add(test1);  
    machine.getTests().add(test2);  
    machine.getTests().add(test5);  
    insert( test1 );  
    insert( test2 );  
    insert( test5 );  
  end
```

Семантические сети

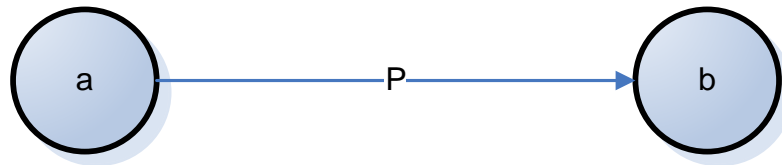
Простые и расширенные семантические сети

Клауза – выражение вида: $B_1, B_2, \dots, B_m \leftarrow A_1, \dots, A_n$
где B_1, B_2, \dots, B_m суть атомарные формулы, $n \geq 0$ и $m \geq 0$.

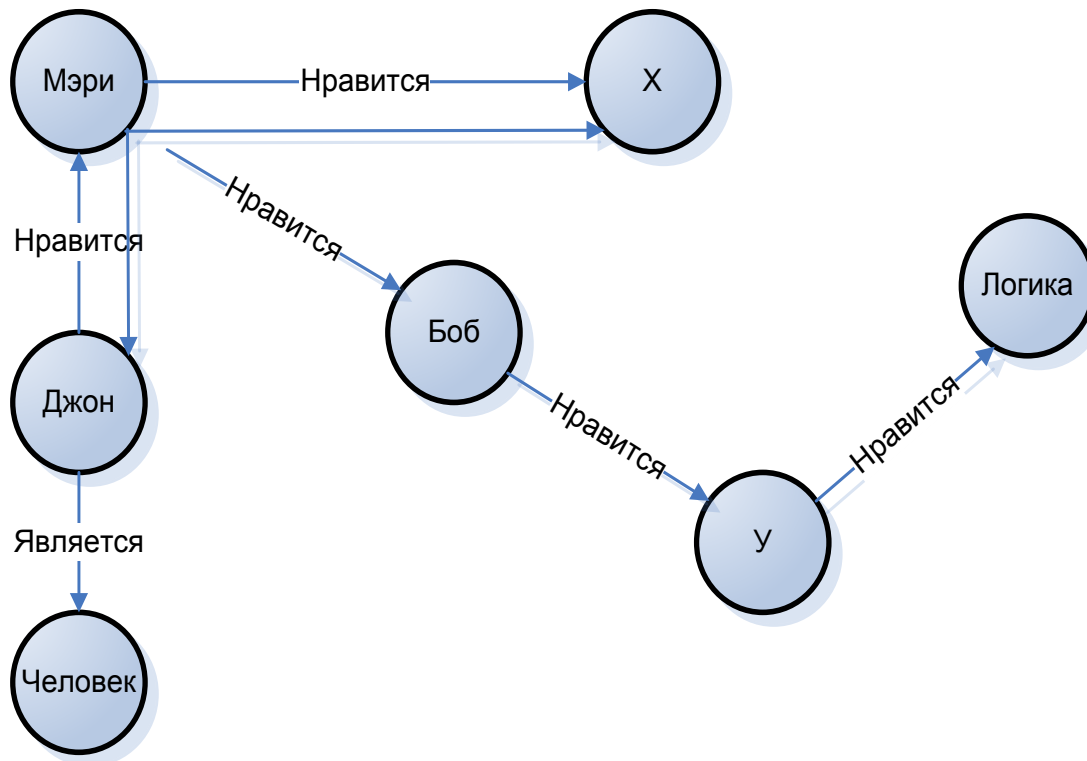
Атомарные формулы A_1, \dots, A_n суть *совместные* посылки клаузы, а B_1, B_2, \dots, B_m суть *альтернативные* заключения. (Множество клауз совместно, если оно истинно в одной из моделей языка).

Простые и расширенные семантические сети

- $P(a,b)$



Простые и расширенные семантические сети



- Джону нравится Мэри ←
- Джон является человеком ←
- Мэри нравится Джон, Мэри нравится Боб ← Мэри нравится X
- Бобу нравится у ← у нравится логика

Универсум Эрбрана и семантические сети

- *Всякий человек, который является хозяином собак, не является хозяином кошек.*
- *Джон является хозяином Линды.*
- *Петя является хозяином Мурки.*

Введем бинарные предикатные символы P – быть хозяином и Q – не быть хозяином.

Универсум Эрбрана и семантические сети

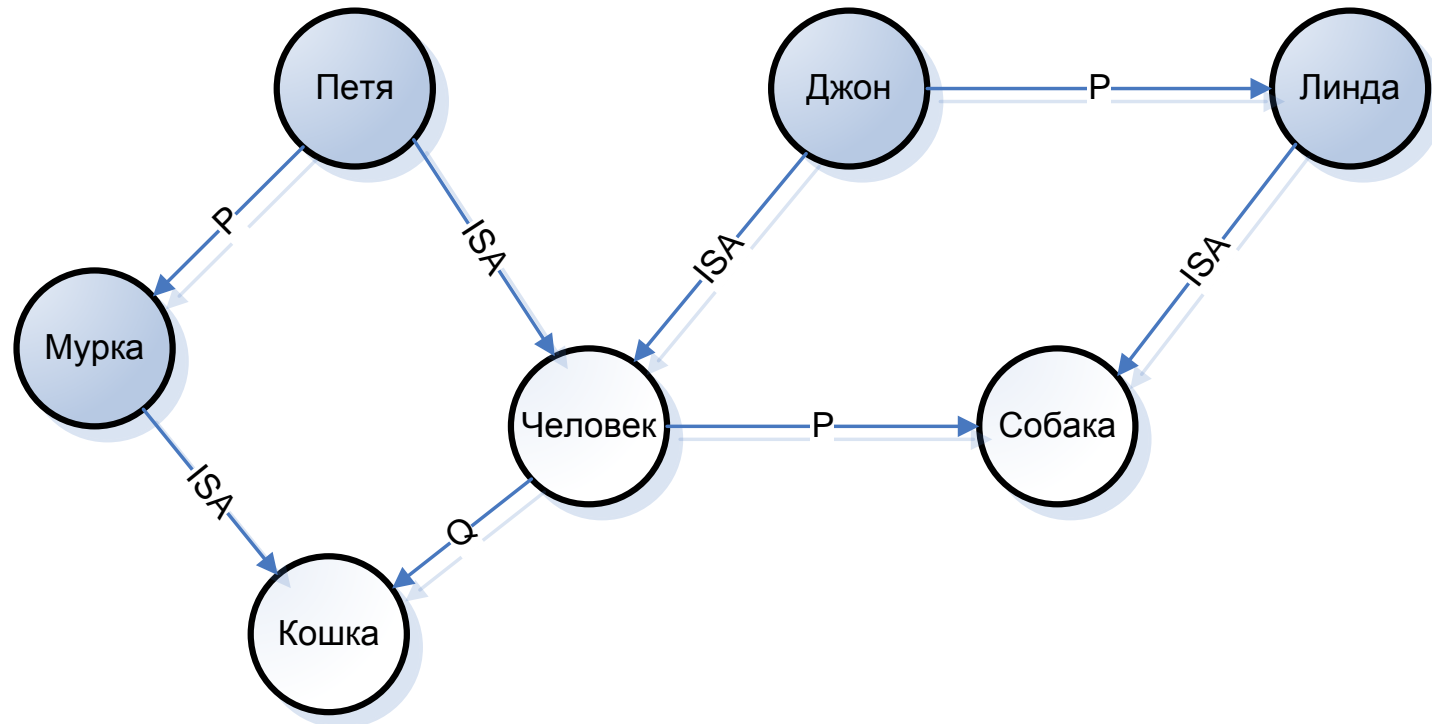
- $Q(\text{человек, кошка}) \leftarrow R(\text{человек, собака})$
- $R(\text{Джон, Линда}) \leftarrow$
- $R(\text{Петя, Мурка}) \leftarrow$

Универсум Эрбрана и семантические сети

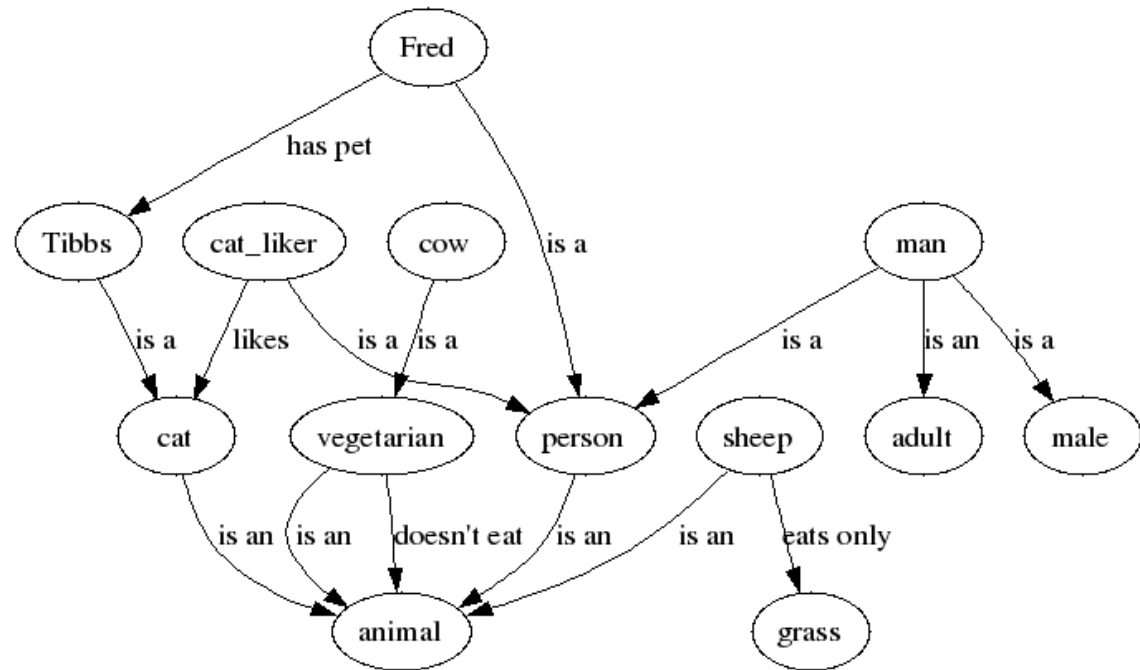
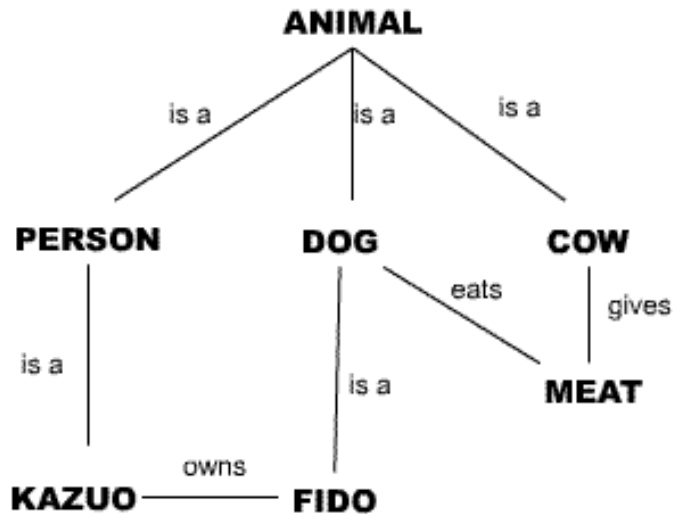
Для полноты картины введем еще один предикатный символ, означающий принадлежность экземпляра общему понятию. В теории интеллектуальных систем его принято обозначать ISA.

- ISA (Джон, человек) ←
- ISA (Петя, человек) ←
- ISA (Линда, собака) ←
- ISA (Мурка, кошка) ←

Универсум Эрбрана и семантические сети



Примеры семантических сетей



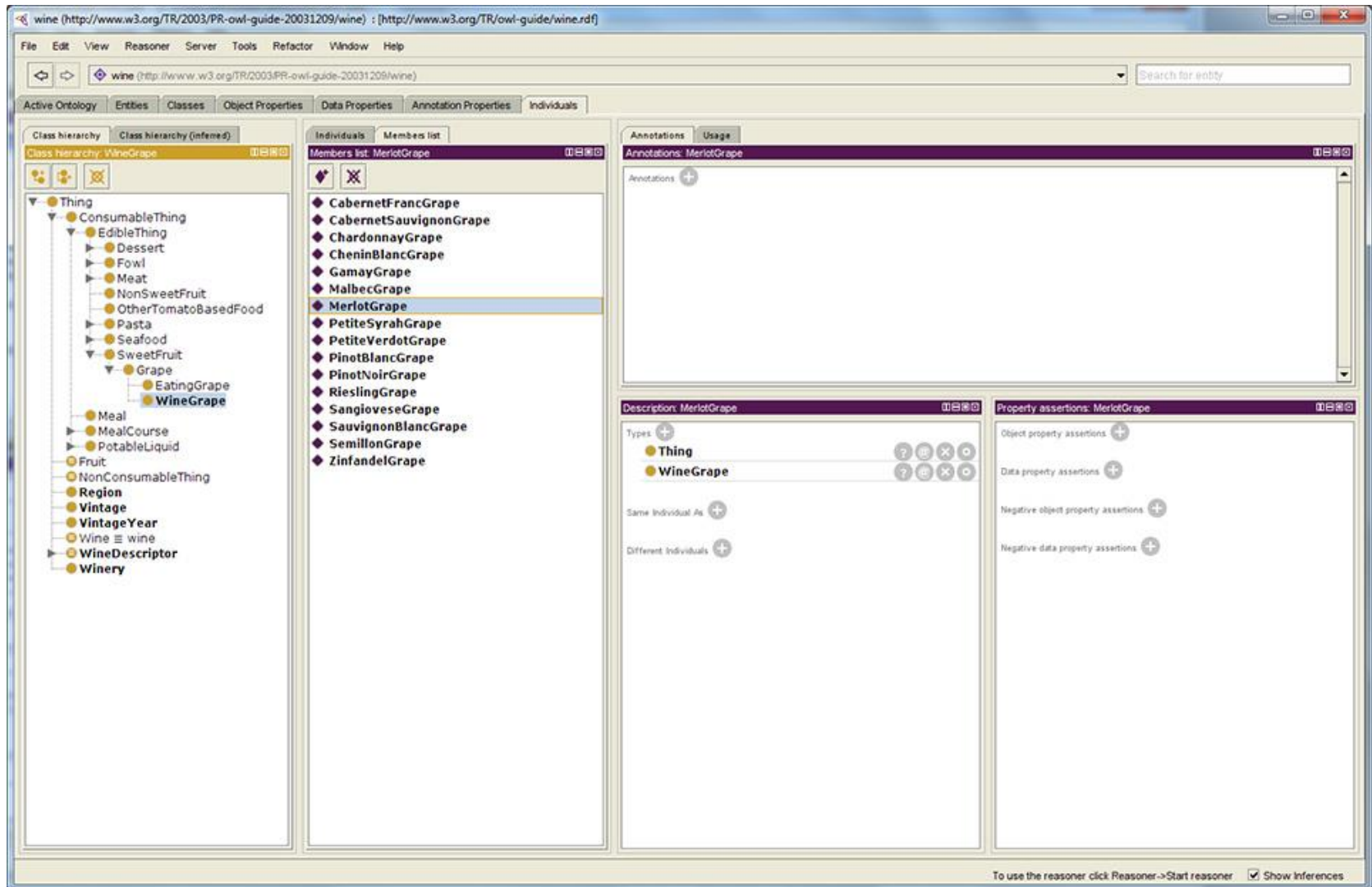
OWL – ЯЗЫК ОПИСАНИЯ ОНТОЛОГИЙ

```
Ontology(  
  Class(pp:male partial)  
  Class(pp:adult partial)  
  Class(pp:elderly partial pp:adult)  
  
  Class(pp:pet complete restriction(pp:is_pet_of someValuesFrom(owl:Thing)))  
  
  Class(pp:animal partial restriction(pp:eats someValuesFrom(owl:Thing)))  
  
  /* Vegetarians do not eat animals or parts of animals */  
  
  Class(pp:vegetarian complete  
    intersectionOf(pp:animal  
      restriction(pp:eats allValuesFrom(complementOf(pp:animal)))  
      restriction(pp:eats  
        allValuesFrom(complementOf(restriction(pp:part_of  
          someValuesFrom(pp:animal)))))))  
  
  DisjointClasses(pp:dog pp:cat)  
  
  ObjectProperty(pp:eaten_by)  
  ObjectProperty(pp:eats inverseOf(pp:eaten_by) domain(pp:animal))  
  
  SubPropertyOf(pp:has_pet pp:likes)  
  
  Individual(pp:Tom type(owl:Thing))  
  Individual(pp:Tibbs type(pp:cat))
```

OWL – язык описания онтологий

```
<owl:Class>
  <owl:intersectionOf rdf:parseType="collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild"/>
      <owl:toClass>
        <owl:unionOf rdf:parseType="collection">
          <owl:Class rdf:about="#Doctor"/>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#hasChild"/>
            <owl:hasClass rdf:resource="#Doctor"/>
          </owl:Restriction>
        </owl:unionOf>
      </owl:toClass>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Редактор онтологий Protege



Редактор онтологий Protege

The screenshot displays the Protege ontology editor interface. The window title is "pizza (http://www.co-ode.org/ontologies/pizza/pizza.owl) : [http://protege.stanford.edu/ontologies/pizza/pizza.owl]". The menu bar includes File, Edit, View, Reasoner, Server, Tools, Refactor, Window, and Help. The address bar shows the URL "pizza (http://www.co-ode.org/ontologies/pizza/pizza.owl)". The interface is divided into several panels:

- Class hierarchy:** A tree view on the left showing the structure of the ontology. The root is "Thing", which branches into "DomainConcept", "Country", "Food", "IceCream", "Pizza", "PizzaBase", "PizzaTopping", and "ValuePartition". The "Pizza" class is expanded, showing subclasses like "CheeseyPizza", "InterestingPizza", "MeatyPizza", "NamedPizza", "NonVegetarianPizza", "RealltalianPizza", "SpicyPizza", "SpicyPizzaEquivalent", "ThinAndCrispyPizza", "VegetarianPizza", "VegetarianPizzaEquivalent1", and "VegetarianPizzaEquivalent2".
- OntoGraf:** The main workspace showing a network graph of classes and their relationships. The graph is centered around the "Pizza" class, which is connected to many other classes. Relationships are represented by colored arrows (purple, green, yellow, orange, red). The graph includes classes like "ThinAndCrispyBase", "Food", "PizzaBase", "Equivalent2", "SpicyPizza", "FruitTopping", "SpicyTopping", "HerbSpiceTopping", "VegetarianPizza", "VegetarianTopping", "InterestingPizza", "MeatTopping", "PizzaTopping", "VegetableTopping", "HotSpicedBeefTopping", "RosemaryTopping", "PeperoniSausageTopping", "ChickenTopping", "HamTopping", and "CajunSpiceTopping".
- Search:** A search bar at the top of the graph area with a dropdown menu set to "contains".
- Footer:** A note at the bottom right says "To use the reasoner click Reasoner->Start reasoner" and a checkbox for "Show Inferences" is checked.

Неоднородные семантические сети

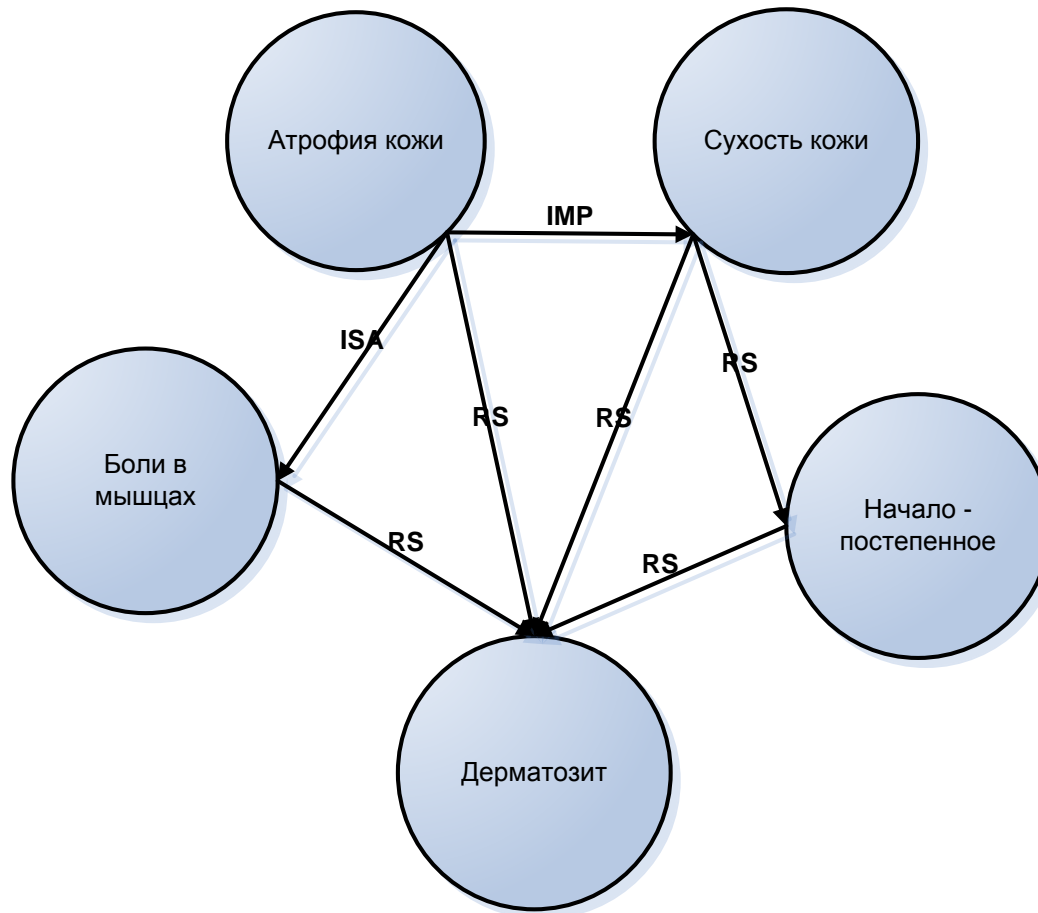
Неоднородная семантическая сеть (НСС) – семейство графов, имеющих общее множество вершин; вершинам сопоставлены объекты моделируемой действительности, ребрам – элементы некоторых бинарных отношений на множестве вершин; ребрам же сопоставлены процедуры, предназначенные для проверки корректности сети и порождения различного рода гипотез, повышающих эффективность процесса построения сети.

Неоднородные семантические сети

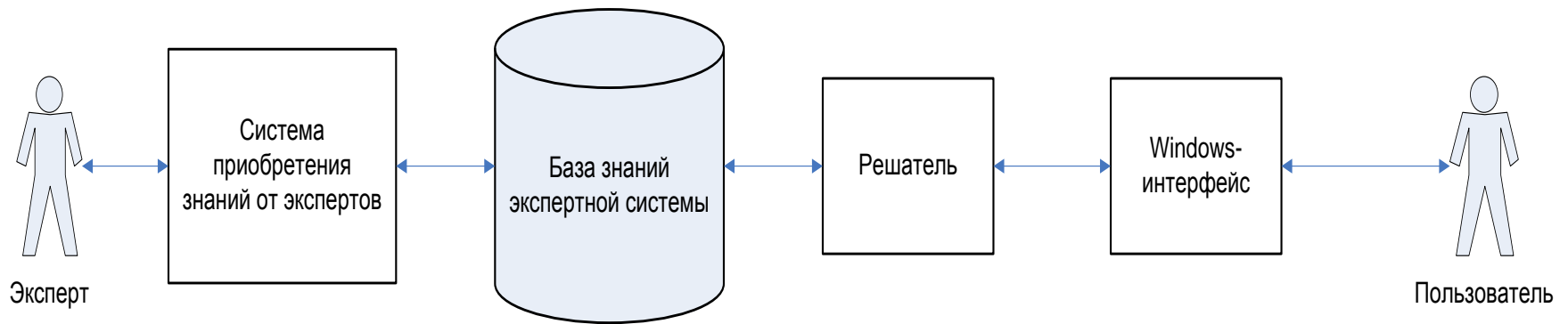
$H = \langle D, N, R, F \rangle$, где

- $D = \{D_1, D_2, \dots, D_n\}$ - семейство непустых множеств;
- $N \subseteq N$ – выделенное подмножество множества слов конечной длины над некоторым алфавитом;
- R - семейство бинарных отношений R_1, R_2, \dots, R_q на N^2 ; $R_i \subseteq N^2$; $i \in \{1, 2, \dots, q\}$.
- $F = \{f_1, f_2, \dots, f_m\}$ семейство функций, каждой из которых приписан некоторый тип. А именно, функция f_i ($i \in \{1, 2, \dots, m\}$) имеет тип $\langle \tau, \omega \rangle$, где $\tau = \langle k_1, k_2, \dots, k_m \rangle$ если она определена на декартовом произведении $D_{k_1} \times D_{k_2} \times \dots \times D_{k_m}$, а областью её значений является множество D_ω , так что каждому кортежу $\delta \in D_{k_1} \times D_{k_2} \times \dots \times D_{k_m}$ функция f_i типа $\langle \tau, \omega \rangle$ из F ставит в соответствие некоторый элемент $f(\delta)$ из D_ω . То обстоятельство, что f_i имеет тип $\langle \tau, \omega \rangle$ будем обозначать $\langle \tau, \omega \rangle (f_i)$.

Пример НСС



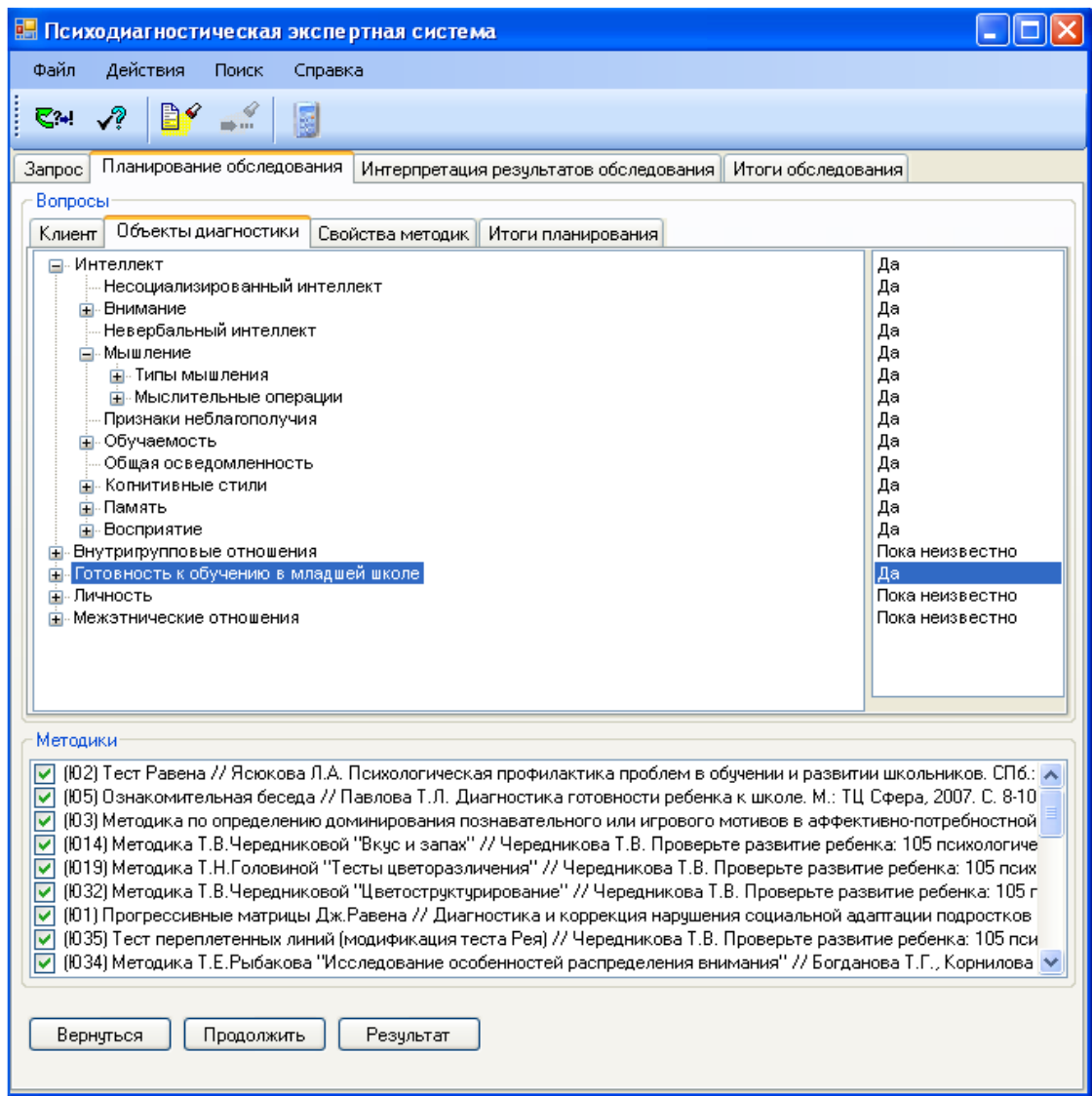
Общая архитектура систем



Алгоритмы приобретения знаний для систем

- Интервью
- Репертуарные решетки
- Извлечение знаний из текстов
- Анализ прецедентов

Примеры систем



Примеры систем

The screenshot shows a software window titled "Экспертная система психологического консультирования" (Expert system for psychological counseling). The window has a menu bar with "Файл", "Действия", "Поиск", and "Справка". Below the menu bar is a tabbed interface with five tabs: "Контакт", "Сбор информации", "Формулировка проблемы", "Влияние", and "Поддержка". The "Контакт" tab is active, displaying a tree view of contact-related issues. The tree view includes "слабый установление контакта" and "нарушения контакта", with the latter expanded to show sub-items like "Фиксация на отношениях, а не на проблеме", "Несовпадение репрезентативных систем", "Ценовая позиция психолога", "Формальные участие", "Нарушение коммуникативных норм (напр., вульгаризмы)", "Тревожность психолога", and "Включение психолога в игровое взаимодействие". To the right of the tree view is a list of possible responses, with "Да" selected. Below the tree view and response list, there is a section titled "Возможная причина" (Possible cause) and "Рекомендуемые действия" (Recommended actions).

Экспертная система психологического консультирования

Файл Действия Поиск Справка

Контакт Сбор информации Формулировка проблемы Влияние Поддержка

- [-] слабый установление контакта
- [-] нарушения контакта
 - Фиксация на отношениях, а не на проблеме
 - Несовпадение репрезентативных систем
 - Ценовая позиция психолога**
 - Формальные участие
 - Нарушение коммуникативных норм (напр., вульгаризмы)
 - Тревожность психолога
 - Включение психолога в игровое взаимодействие

Пска неизвестно
Пска неизвестно
Пска неизвестно
Пска неизвестно
Да
Пска неизвестно
Пска неизвестно
Пска неизвестно
Пска неизвестно

Возможная причина

- Стремление психолога навязать свою систему ценностей клиенту
- Стремление психолога продемонстрировать превосходство над клиентом
- Неверие психолога в разрешимость проблемы ситуации консультирования

Рекомендуемые действия

- Рефлексия потребности в доминировании в ситуации консультирования

Примеры систем

Боль в спине

Файл Поиск Окно Действия Справка

Данные **Обследование** Результат

Жалобы **Данные клинических исслед...** Данные рентгенологических исс... Анамнез Компьютерная томография

Ассимметричное положение надплечий и лопаток	Пока неизвестно
Ассимметрия грудных желез	Пока неизвестно
Ассимметрия пупка	Пока неизвестно
Атоничность и дряблость поврежденной связки	Пока неизвестно
Атрофия кожи	Пока неизвестно
Атрофия мышц ног	Пока неизвестно
Болезненность в области матки, придатков	Пока неизвестно
Болезненность в эпигастриальной области при пальпации	Пока неизвестно
Болезненность при пальпации в проекции нервных стволов	Пока неизвестно
Болезненность при пальпации и перкуссии остистого отростка в зоне поражения	Пока неизвестно
Болезненность при пальпации крестцово-подвздошного сочленения	Пока неизвестно
Болезненность при пальпации мышц плечевого и тазового пояса	Пока неизвестно
Болезненность при пальпации надостной и межостистой связок	Да
Болезненность при пальпации операционного рубца	Пока неизвестно
Болезненность при пальпации остистого отростка в зоне операционного рубца	Пока неизвестно
Болезненность при пальпации остистого отростка пораженного позвонка	Да
Болезненность при пальпации остистых отростков	Пока неизвестно
Болезненность при пальпации остистых отростков Th12-L1 позвонков	Пока неизвестно
Болезненность при пальпации остистых отростков в грудном отделе позвоночника	Пока неизвестно
Болезненность при пальпации остистых отростков и паравертебральных точек на уровне С	Пока неизвестно
Болезненность при пальпации остистых отростков и паравертебральных точек на уровне L	Пока неизвестно
Болезненность при пальпации остистых отростков на вершине искривления	Пока неизвестно
Болезненность при пальпации остистых отростков поврежденных и смежных позвонков	Пока неизвестно
Болезненность при пальпации паравертебральной области в проекции поврежденных позвонков	Пока неизвестно
Болезненность при пальпации паравертебральных точек	Да
Болезненность при пальпации паравертебральных точек в грудном отделе позвоночника	Пока неизвестно
Болезненность при пальпации паравертебральных точек на вершине искривления	Пока неизвестно
Болезненность при пальпации остистого отростка смещенного позвонка	Пока неизвестно
Боли по ходу периферических (чаще - межреберных) нервов	Пока неизвестно

Гипотезы

- Повреждения надостной и межостистой связок
- Дерматомиозит
- Дегенеративная болезнь шейного отдела позвоночника с синдромом затылочного нерва
- Переломы поперечных отростков позвонков
- Болезнь Колье
- Диспластический сколиоз поясничного отдела позвоночника
- Врожденный сколиоз
- Синдром неудачно оперированного позвоночника
- Спондилолиз
- Синдром Клиппеля-Фейля

В разделе = Данные клинических исследований Гипотез = 11 Объектов = 218 Всего = 218

Фреймы

Представление знаний в системах фреймов

1. Именованные слоты, которые могут заполняться данными, например, слот ЧИСЛО КОНЕЧНОСТЕЙ во фрейме ЧЕТВЕРОНОГОЕ ЖИВОТНОЕ. Данные, заполняющие слот, могут быть строкового типа, целого, булевского и т.д. Некоторые слоты могут заполняться по умолчанию. Например, в том же фрейме, слот НАЛИЧИЕ ШЕРСТИ может заполняться по умолчанию, т.к. в большинстве случаев, это справедливо.

2. Слоты могут иметь тип ISA или АКО. Слот ISA указывает на участие рассматриваемого фрейма в иерархии фреймов и содержит имя фрейма, соответствующего большему классу; например, для фрейма ЧЕТВЕРОНОГОЕ ЖИВОТНОЕ, это может быть фрейм ЖИВОТНОЕ.

Слот АКО указывает на родовую принадлежность фрейма, т.е. на наличие у него родового или видового свойства; например, для фрейма ЖИВОТНОЕ это может быть свойство МЛЕКОПИТАЮЩЕЕ. При этом это свойство может наследоваться фреймом ЧЕТВЕРОНОГОЕ ЖИВОТНОЕ по ISA – иерархии.

3. Слоты могут носить процедурный характер. Например, во фрейме СОБАКА, значение слота КОЛИЧЕСТВО ЕЖЕДНЕВНО ПОТРЕБЛЯЕМОЙ ПИЩИ вычисляется как функция ее размера, веса и возраста. Разумеется, фрейм должен содержать соответствующие слоты, а именно, ВЕС, РАЗМЕР, ВОЗРАСТ.

Представление знаний в системах фреймов

определение фрейма в *нотации Бэкуса-Наура*:

- $\langle \text{фрейм} \rangle ::= \langle \text{имя фрейма} \rangle \{ \langle \text{тело фрейма} \rangle \}$
- $\langle \text{тело фрейма} \rangle ::= \langle \text{множество слотов} \rangle$
- $\langle \text{множество слотов} \rangle ::= \langle \text{слот} \rangle \mid \langle \text{слот} \rangle, \langle \text{множество слотов} \rangle$
- $\langle \text{слот} \rangle ::= \langle \text{имя слота} \rangle : \langle \text{значение слота} \rangle$
- $\langle \text{значение слота} \rangle ::= \langle \text{имя фрейма} \rangle \mid \langle \text{имя процедуры} \rangle \mid \langle \text{множество} \rangle$
- $\langle \text{множество} \rangle ::= \langle \text{дискретное множество} \rangle \mid \langle \text{плотное множество} \rangle$
- $\langle \text{дискретное множество} \rangle ::= \langle \text{элемент множества} \rangle, \langle \text{множество} \rangle$
- $\langle \text{элемент множества} \rangle ::= \langle \text{идентификатор} \rangle$
- $\langle \text{плотное множество} \rangle ::= \langle \text{интервал} \rangle \mid \langle \text{полуинтервал} \rangle \mid \langle \text{отрезок} \rangle$

Примеры фреймов

Mammal:

subclass: Animal
warm-blooded: yes
*furry: yes

Elephant:

subclass: Mammal
has-trunk: yes
*colour: grey
*size: large
*furry: no

Nellie:

instance: Elephant
colour: pink
owner: Fred
size: small

Elephant:

subclass: Mammal
has-trunk: yes
colour: grey
size: large
habitat: jungle

Nellie:

instance: Elephant

Описание фрейма с помощью Lisp

Can represent frames as association lists in the form
defvar frame

```
`((mammal (limbs, 4))  
  (elephant (subclass . mammal)  
            (colour . grey))  
  (nellie   (instance . elephant)  
            (tusks . 2)))
```

Or sets of tuples)

```
`((limbs mammal 4)  
  (subclass elephant mammal)  
  (colour elephant grey)  
  (instance nellie elephant)  
  (tusks nellie 2))
```

Спасибо за внимание!