Лекция 12. Представление знаний и экспертные системы.
Экспертная система компьютерного познания.

«Логический подход к искусственному интеллекту»

1. Концептуальный граф. Представляет логическую формулу.

Имена и аргументы предикатов представляются двумя типами узлов.

Дуги графа соединяют имена предикатов с их аргументами.

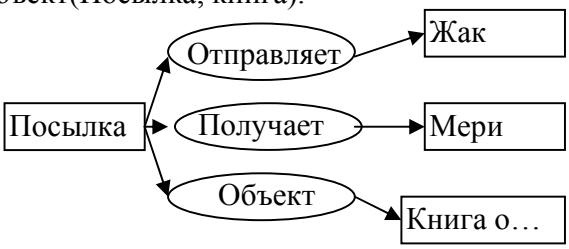
Содержат прямоугольники для представления аргументов и круги для представления имен предикатов.

Представим фразу: «Жак посылает книгу Мери»

Отправитель(Посылка, Жак);

Получатель(Посылка, Мери);

Объект(Посылка, книга).



2. Семантические сети. Семантические сети получаются из концептуальных графов по правилам соединения:

Правило конъюнкции. Если узел-концепт одной сети/графа идентичен узелу-концепту другой сети/графа, то эти сети/графы можно объединить совместив узелы-концепты.

Правило упрощения. Если сеть содержит идентичные узлы, связывающие одни те же узлы-концепты, то один из узлов можно удалить.

Представление контекста.

Это(Книга о…, книга);

Элем(Книга о…, библиотека ун);

Подм(книга_ЭС, книга_ИИ).

3. Фреймы. объектное представление.

Бинарная версия исчисления предикатов привела к семантическим сетям.

Аналогично объекты представляются тройками:

(объект, атрибут_i, значение_j)

Сцепки:

Жак_2

Пишет(Жак_2, Книга_22);

Посылает(Жак_2, Мари_4; Книга_22)

Фреймы – сцепки, представленные бинарными предикатами.

Концепту Посылает, представленному предикатом Посылает() соответствует фрейм, если представить этот предикат произведением бинарных предикатов:

Отправитель(Посылает, Жак_2)&

Получатель(Посылает, Мери_4)&

Объект(Посылает, Книга_22)

Фрейм Посылает:		(объект)
Отправитель	Жак_2	(слот-1)
Получатель	Мери_4	(слот-2)
Объект	Книга_22	(слот-3)


ОНТОЛОГИЯ, Предмет исследования
Система понятий, признаки, величины. Данные – разных типов, взятые из баз данных.
Априорные знания, экспертные знания, предпочтения лиц принимающих решения.
КОЛИЧЕСТВЕННАЯ ЭМПИРИЧЕСКАЯ ТЕОРИЯ
вещественные числа
Данные, экспертные оценки – массивы и матрицы числовых значений величин.
⇒ Извлечение из признаков, величин интерпретируемой информации и представление числовых данных многосортными эмпирическими системами;
⇐ Аксиоматический анализ величин и определение шкал;
Априорные, экспертные знания;
Индуктивные знания – числовые представления функциональных зависимостей, классов, сплайнов, решающих правил, деревьев и т.д;
⇒ аксиоматический анализ зависимостей.
⇐ определение новых шкал величин, упрощение функциональных зависимостей перешкалированием величин; Обнаружение законов.
ЛОГИЧЕСКАЯ ЭМПИРИЧЕСКАЯ ТЕОРИЯ
Процедуры извлечения интерпретируемой информации из данных;
Данные – многосортная эмпирическая система M;
Определение групп допустимых преобразований шкал.
Априорные знания – система аксиом, исходная теория ПО Th(M)
Индуктивные знания – семантический вероятностный вывод множеств законов L, LP, MSR
Множество законов L ⊢ Th(M).
Множество вероятностных законов LP.
Множество максимально специфических вероятностных законов MSR.
Аппроксимация дедуктивного вывода семантическим выводом.
Экспертные знания.
КОНСТРУКТИВНАЯ ЭМПИРИЧЕСКАЯ ТЕОРИЯ рациональные и натуральные
Данные, экспертные оценки, предпочтения лиц принимающих решения – эффективно вычислимые числовые представления структурных величин, порядков, решеток, предпочтений и т. д.;
Априорные, экспертные знания в виде анкет тестов …
Индуктивные знания – конструктивное шкалирование экспертных предпочтений, функций полезности;
Конструктивные зависимости, анкеты, тесты.
Конструктивное представление величин.
ПОЛЬЗОВАТЕЛЬ. ВОПРОСНО-ОТВЕТНАЯ СИСТЕМА.
Вывод предсказаний, решений, диагнозов из экспертной системы.

Научно-исследовательская экспертная система.
Система компьютерного познания.

Для этого надо сначала задать предметную область. Задание предметной области осуществляется заданием онтологии. Поэтому первый шаг в обнаружении эмпирических теорий состоит в задании онтологии.

Онтология включает:

- систему понятий ПО, в которой формулируется и интерпретируется эмпирическая теория;
- свойства, признаки, величины и соответствующие измерительные процедуры, интерпретируемые в системе понятий;
- априорные и экспертные знания;
- знания, интерпретируемые в системе понятий ПО, получаемые в процессе построения логической, количественной и конструктивной эмпирических теорий.

Мы предполагаем онтологию заданной.

1. Построение логической эмпирической теории (ЛЭТ)**.** Для ее построения необходимо выделить логико-операционную составляющую чисел и данных в соответствии с методологическим принципом теории измерений: «свойства определяются отношениями». Для этого надо решить задачу.

Задача 1. Определить множество Ω интерпретируемых в системе понятий ПО отношений и операций для всех понятий, свойств, величин, и признаков онтологии и представить данные в виде многосортной эмпирической системы.

Для решения этой задачи вводится понятие «эмпирическое содержание данных», формализуемое с помощью эмпирической теорией. Было показано, как такие известные типы данных, как парные и множественные сравнения, матрицы упорядочений, матрицы близости и матрицы объект–признак могут быть представлены в эмпирических аксиоматических теориях многосортными эмпирическими системами. Были приведены результаты теории измерений, относящиеся к соответствующим отношениям и операциям. Для отношений и операций можно найти множество законов, из которых выводится теория ПО.

Априорные знания онтологии также нужно представить системой аксиом S^{Ω}, представленной совокупностью универсальных формул..

Экспертные знания могут быть извлечены из эксперта описанным ранее методом, основанном на монотонных булевых функциях.

Задача 2. Разработать индуктивный метод обнаружения закономерностей вида **Error! Reference source not found.** на данных, представленных многосортными эмпирическими системами.

Такой метод основан на семантическом вероятностном выводе и обладает целым рядом важных свойств. Полученная в результате применения метода логическая эмпирическая теория также обладает целым рядом важных свойств.

2. Построение количественной эмпирической теории (КЭТ) осуществляется на основании результатов теории измерений, дающих числовые представления величин / законов. В теории измерений найдены системы аксиом для многих физических величин и фундаментальных физических законов. Если в ЛЭТ содержится какая-либо система аксиом теории измерений, то она дает числовые представления величин и функциональных зависимостей. Эти числовые представления величин и функциональных зависимостей, получаемые из систем аксиом, интерпретируемы в системе понятий ПО.

Проблема в построении КЭТ состоит в том, что далеко не для всех систем аксиом, которые могут быть получены в результате индуктивного вывода ЛЭТ, существуют соответствующие им результаты теории измерений.

Задача 3. Определить классификацию всех возможных законов природы.

Единственной теорией, в которой такая классификация существует, является теория физических структур (ТФС). Была приведена классификация возможных законов природы, полученная в ТФС. Нами установлена связь между ТФС и теорией измерений.

Задача 4. Найти обобщение теории измерений, которое бы позволяло строить числовые представления величин и законов практически для любой системы аксиом.

Такое обобщение получено путем использования теории конструктивных моделей. Значениями величин в этом случае являются натуральные, рациональные или другие эффективно вычислимые числа (например, коды). Теория конструктивных моделей наиболее полно отражает смысл построения числовых представлений – закодировать эмпирическую систему числами или кодами так, чтобы можно было легко и удобно по этим кодам вычислять все значения отношений и операций на эмпирической системе. В результате такой кодировки мы получаем эмпирическую теорию, которую мы назвали конструктивной.

Таким образом, по обнаруженной системе аксиом строятся либо числовые, либо конструктивные числовые представления.

3. Построение конструктивной эмпирической теории (КонЭТ)**.** В теории измерений нельзя получить числовые представления некоторых величин и законов в силу ограниченности понятия числового представления. Величины и законы, описываемые частичными порядками, толерантностями, решетками и т. д., не могут быть сильным гомоморфизмом вложены в поле вещественных чисел. Для числового представления таких величин и закономерных связей нами предложено использовать конструктивные числовые представления.

Примерами конструктивных числовых представлений законов являются, например, психологические тесты.

4. Цикл познания в теорией измерений. Таким образом, нами разработаны понятия и методы, позволяющие осуществлять следующий цикл познания, обозначенный на рисунке двойными стрелками:

- определить онтологию предметной области;
- извлечь из числовых представлений величин множества отношений и операций, определяющие смысл этих величин в соответствии с теорией измерений. Перевести данные в многосортные эмпирические системы, используя найденные множества отношений и операций. Перевести априорные и экспертные знания в формулы вида (1);
- определить системы аксиом, которым удовлетворяют величины и законы;
- найти числовые представления величин и законов в теории измерений и / или в теории конструктивных моделей;
- проинтерпретировать полученные числовые представления в системе понятий онтологи и получить в результате систему величин связанных между собой законами как это имеет место в физике.

Объяснение и работа с оценками в экспертных системах. Метаинтерпретаторы.

Программа, которая в случае нехватки информации, запрашивает информацию у пользователя.

```
solve(Goal) ←

solve(true).
solve((A,B)) ← solve(A), solve(B).
solve(A) ← clause(A,B), solve(B).
solve(A) ← askable(A), not known(A), ask(A,Answer), respond(Answer,A).
```

первоначально всегда not known(A)
askable – процедура, коротая в случае безуспешного решения цели интерпретатором
может направить ее на рассмотрение пользователю.
ask(A, Answer) ← display_query(A), read(Answer).
respond(yes, A) ← assert(A) – в программу вводится факт A.
respond(no, A) ← assert(untrue(A)), fail. (предикат, который никогда не выполняется).
known(A) ← A.
known(A) ← untrue(A).
display_query(A) ← write(A), write(?).

Программа, объясняющая как доказывается цель.
how(Goal) ←

how(Goal) ← solve(Goal, Proof), interpret(Proof).

solve(true, true).
solve((A,B),(ProofA, ProofB)) ← solve(A, ProofA), solve(B, ProofB).
solve(A, (A ← ProofB)) ← clause(A, B), solve(B, ProofB).

interpret((Proof1, Proof2)) ← interpret(Proof1), interpret(Proof2).
interpret(Proof) ← fact(Proof, Fact), nl, writeln([Fact,'факт в базе данных'])
fact((Fact ← true), Fact).
interpret(Proof) ← rule(Proof, Head, Body, Proof1),
nl, writeln([Head, 'доказывается с помощью правила']).
display_rule(rule(Head, Body)),
interpret(Proof1).

rule((Goal ← Proof), Goal, Body, Proof) ← Proof ≠ true, extract_body(Proof, Body).

extract_body((Proof1, Proof2), (Body1, Body2)) ←
extract_body(Proof1, Body1), extract_body(Proof2, Body2).
extract_body((Goal ← Proof), Goal).
display_rule(rule(A, B)) ← write('IF'), wreite_conjunction(B), writeln(['THEN', A])

Программа вычисления оценок утверждений.
solve(Goal, Certainty) ←

solve(true, 1).
solve((A,B), C) ← solve(A, C1), solve(B, C2), minimum(C1,C2,C).
solve(A, C) ← clause_cf(A, B, C1), solve(B, C2), C := C1*C2.
clause_cf(A, B, C1) – предикат дающий оценку определенности правила A ← B.