



Федеральное агентство по рыболовству
БГАРФ ФГБОУ ВО «КГТУ»
Калининградский морской рыбопромышленный колледж

Утверждаю
Заместитель начальника колледжа
по учебно-методической работе

А.И.Колесниченко

МДК.06.04 Интеллектуальные системы и технологии

Методические указания для выполнения практических занятий
по специальности

09.02.07 Информационные системы и программирование

МО-09 02 07-МДК.06.04.ПЗ

РАЗРАБОТЧИКИ Богатырева Т.Н.

ЗАВЕДУЮЩИЙ ОТДЕЛЕНИЕМ Бакулин А.М.

ГОД РАЗРАБОТКИ 2022

ГОД ОБНОВЛЕНИЯ 2025

МО-09 02 07-МДК.06.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ТЕХНОЛОГИИ	С. 2/25

МО-09 02 07-МДК.06.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ТЕХНОЛОГИИ	С. 3/25

Содержание

Практическое занятие №1 Моделирование интеллектуальных систем. Классификация интеллектуальных информационных систем	4
Практическое занятие №2 Моделирование интеллектуальных систем. Логическая модель представления знаний	9
Практическое занятие №3 Моделирование интеллектуальных систем. Предикаты ..	17
Практическое занятие №4 Моделирование интеллектуальных систем. Продукционная модель представления знаний.....	17
Практическое занятие №5 Моделирование интеллектуальных систем. Семантические ИС.....	22

Практическое занятие №1 Моделирование интеллектуальных систем. Классификация интеллектуальных информационных систем

Содержание и порядок выполнения задания:

Изучите теоретический материал

Выполните задания.

Классификация интеллектуальных информационных систем

Определение интеллектуальной информационной системы

Существует большое множество интеллектуальных информационных систем (ИИС). Однако общепринятого единого определения интеллектуальной информационной системы нет.

Интеллектуальной информационной системой называют автоматизированную информационную систему, основанную на знаниях, или комплекс программных, лингвистических и логико-математических средств для реализации основной задачи – осуществления поддержки деятельности человека и поиска информации в режиме продвинутого диалога на естественном языке.

Кроме того, информационно-вычислительными системами с интеллектуальной поддержкой для решения сложных задач называют те системы, в которых логическая обработка информации превалирует над вычислительной.

Таким образом, любая информационная система, решающая интеллектуальную задачу или использующая методы искусственного интеллекта, относится к интеллектуальным.

Для интеллектуальных информационных систем характерны следующие признаки:

- развитые коммуникативные способности;
- умение решать сложные плохо формализуемые задачи;
- способность к самообучению;
- адаптивность.

Коммуникативные способности ИИС характеризуют способ взаимодействия (интерфейса) конечного пользователя с системой, в частности возможность формулирования произвольного запроса в диалоге с ИИС на языке, максимально приближенном к естественному.

Сложные плохо формализуемые задачи – это задачи, которые требуют построения оригинального алгоритма решения в зависимости от конкретной ситуации, для которой могут быть характерны неопределенность и динамичность исходных данных и знаний.

*Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж*

Способность к самообучению – это возможность автоматического извлечения знаний для решения задач из накопленного опыта конкретных ситуаций.

Адаптивность – способность к развитию системы в соответствии с объективными изменениями модели проблемной области.

Классификация интеллектуальных систем

В соответствии с перечисленными признаками ИИС делятся на (данная классификация – одна из возможных) (рис. 1):

- системы с коммутативными способностями (с интеллектуальным интерфейсом);
- экспертные системы (системы для решения сложных задач);
- самообучающиеся системы (системы, способные к самообучению);
- адаптивные системы (адаптивные информационные системы).

Классификация интеллектуальных информационных систем по типам систем

Системы с коммутативными способностями

Интеллектуальные базы данных отличаются от обычных баз данных возможностью выборки по запросу необходимой информации, которая может явно не храниться, а выводиться из имеющейся в базе данных.

Естественно-языковой интерфейс предполагает трансляцию естественно-языковых конструкций на внутримашинный уровень представления знаний. Для этого необходимо решать задачи морфологического, синтаксического и семантического анализа и синтеза высказываний на естественном языке. Так, морфологический анализ предполагает распознавание и проверку правильности написания слов по словарям, синтаксический контроль – разложение входных сообщений на отдельные компоненты (определение структуры) с проверкой соответствия грамматическим правилам внутреннего представления знаний и выявления недостающих частей и, наконец, семантический анализ – установление смысловой правильности синтаксических конструкций. Синтез высказываний решает обратную задачу преобразования внутреннего представления информации в естественно-языковое.

Естественно-языковой интерфейс используется для:

- доступа к интеллектуальным базам данных;
- контекстного поиска документальной текстовой информации;
- голосового ввода команд в системах управления;

- машинного перевода с иностранных языков.

Гипертекстовые системы предназначены для реализации поиска, по ключевым словам, в базах текстовой информации. Интеллектуальные гипертекстовые системы отличаются возможностью более сложной семантической организации ключевых слов, которая отражает различные смысловые отношения терминов. Таким образом, механизм поиска работает прежде всего с базой знаний ключевых слов, а уже затем непосредственно с текстом. В более широком плане сказанное распространяется и на поиск мультимедийной информации, включающей, помимо текстовой, и цифровую информацию.

Системы контекстной помощи можно рассматривать как частный случай интеллектуальных гипертекстовых и естественно-языковых систем. В отличие от обычных систем помощи, навязывающих пользователю схему поиска требуемой информации, в системах контекстной помощи пользователь описывает проблему (ситуацию), а система с помощью дополнительного диалога ее конкретизирует, и сама выполняет поиск относящихся к ситуации рекомендаций. Такие системы относятся к классу систем распространения знаний (KnowledgePublishing) и создаются как приложение к системам документации (например, технической документации по эксплуатации товаров).

Системы когнитивной графики позволяют осуществлять интерфейс пользователя с ИИС с помощью графических образов, которые генерируются в соответствии с происходящими событиями. Такие системы используются в мониторинге и управлении оперативными процессами. Графические образы в наглядном и интегрированном виде описывают множество параметров изучаемой ситуации. Например, состояние сложного управляемого объекта отображается в виде человеческого лица, на котором каждая черта отвечает за какой-либо параметр, а общее выражение лица дает интегрированную характеристику ситуации. Системы когнитивной графики широко используются также в обучающих и тренажерных системах на основе использования принципов виртуальной реальности, когда графические образы моделируют ситуации, в которых обучаемому необходимо принимать решения и выполнять определенные действия.

Экспертные системы

Экспертные системы предназначены для решения задач на основе накапливаемой базы знаний, отражающей опыт работы экспертов в рассматриваемой проблемной области.

Многоагентные системы. Для таких динамических систем характерна интеграция в базе знаний нескольких разнородных источников знаний,

обменивающихся между собой получаемыми результатами на динамической основе.

Для *многоагентных систем* характерны следующие особенности:

- 1) проведение альтернативных рассуждений на основе использования различных источников знаний с механизмом устранения противоречий;
- 2) распределенное решение проблем, которые разбиваются на параллельно решаемые подпроблемы, соответствующие самостоятельным источникам знаний;
- 3) применение множества стратегий работы механизма вывода заключений в зависимости от типа решаемой проблемы;
- 4) обработка больших массивов данных, содержащихся в базе данных;
- 5) использование различных математических моделей и внешних процедур, хранимых в базе моделей;
- 6) способность прерывания решения задач в связи с необходимостью получения дополнительных данных и знаний от пользователей, моделей, параллельно решаемых подпроблем.

Самообучающиеся системы

В основе *самообучающихся систем* лежат методы автоматической классификации примеров ситуаций реальной практики.

Характерными признаками самообучающихся систем являются:

- самообучающиеся системы «с учителем», когда для каждого примера задается в явном виде значение признака его принадлежности некоторому классу ситуаций (классообразующего признака);
- самообучающиеся системы «без учителя», когда по степени близости значений признаков классификации система сама выделяет классы ситуаций.

Индуктивные системы используют обобщение примеров по принципу от частного к общему. Процесс классификации примеров осуществляется следующим образом:

1. Выбирается признак классификации из множества заданных (либо последовательно, либо по какому-либо правилу, например, в соответствии с максимальным числом получаемых подмножеств примеров).
2. По значению выбранного признака множество примеров разбивается на подмножества.
3. Выполняется проверка, принадлежит ли каждое образовавшееся подмножество примеров одному подклассу.

4. Если какое-то подмножество примеров принадлежит одному подклассу, то есть у всех примеров подмножества совпадает значение классообразующего признака, то процесс классификации заканчивается (при этом остальные признаки классификации не рассматриваются).

5. Для подмножеств примеров с несовпадающим значением классообразующего признака процесс классификации продолжается, начиная с пункта 1 (каждое подмножество примеров становится классифицируемым множеством).

Нейронные сети представляют собой устройства параллельных вычислений, состоящие из множества взаимодействующих простых процессоров. Каждый процессор такой сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам.

В экспертных системах, *основанных на прецедентах* (аналогиях), база знаний содержит описания не обобщенных ситуаций, а собственно сами ситуации или прецеденты.

Поиск решения проблемы в экспертных системах, основанных на прецедентах, сводится к поиску по аналогии (то есть абдуктивный вывод от частного к частному).

В отличие от интеллектуальной базы данных, *информационное хранилище* представляет собой хранилище извлеченной значимой информации из оперативной базы данных, которое предназначено для оперативного ситуационного анализа данных (реализации OLAP- технологии).

Типичными задачами оперативного ситуационного анализа являются:

- определение профиля потребителей конкретных объектов хранения;
- предсказание изменений объектов хранения во времени;
- анализ зависимостей признаков ситуаций (корреляционный анализ).

Адаптивные системы

Адаптивная информационная система – это информационная система, которая изменяет свою структуру в соответствии с изменением модели проблемной области.

При этом:

1. адаптивная информационная система должна в каждый момент времени адекватно поддерживать организацию бизнес-процессов;
2. адаптивная информационная система должна проводить адаптацию всякий раз, как возникает потребность в реорганизации бизнес- процессов;
3. реконструкция информационной системы должна проводиться быстро и с

МО-09 02 07-МДК.06.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ТЕХНОЛОГИИ	С. 9/25

минимальными затратами.

Ядром адаптивной информационной системы является постоянно развиваемая модель проблемной области (предприятия), поддерживаемая в специальной базе знаний – репозитории. На основе ядра осуществляется генерация или конфигурация программного обеспечения. Таким образом, проектирование и адаптация ИС сводится, прежде всего, к построению модели проблемной области и ее своевременной корректировке.

Так как нет общепринятого определения, четкую единую классификацию интеллектуальных информационных систем дать затруднительно.

Задание

Создать блок-схему для классификации ИИС по типам

Практическое занятие №2 Моделирование интеллектуальных систем. Логическая модель представления знаний

Логическая модель основана на системе исчисления предикатов первого порядка. Знакомство с логикой предикатов начнем с исчисления высказываний.

Высказыванием называется предложение, относительно которого имеет смысл утверждать истинно (Т) оно или ложно (F). Например, предложения «лебедь белый» и «лебедь черный» будут высказываниями. Из простых высказываний можно составить более сложные:

«лебедь белый или лебедь черный»,

«лебедь белый и лебедь черный»,

«если лебедь не белый, то лебедь чёрный».

В свою очередь, сложные высказывания можно разделить на простые, которые связаны между собой с помощью слов: и, или, не, если — то. Элементарными (простыми) называются высказывания, которые нельзя разделить на части. Логика высказываний оперирует логическими связями между высказываниями, т. е. она решает вопросы типа: «Можно ли на основе высказывания А получить высказывание В?»; «Истинно ли В при истинности А?» и т.п. При этом семантика высказываний не имеет значения. Элементарные высказывания рассматриваются как

переменные логического типа, над которыми разрешены следующие логические операции:

- \neg отрицание (унарная операция);
- \wedge конъюнкция (логическое умножение);
- \vee дизъюнкция (логическое сложение);
- \rightarrow импликация (если — то);
- \leftrightarrow эквиваленция.

Операция импликации должна удовлетворять следующим требованиям.

1. Значение результата импликации зависит от двух операндов.
2. Если первый операнд (A) — истинный, то значение результата совпадает со значением второго операнда (B).
3. Операция импликации не коммутативна.
4. Результат импликации совпадает с результатом выражения $\bar{A} \vee B$.

Значения результатов логических операций над переменными X и Y, являющимися элементарными высказываниями, приведены в табл. 5.1.

Таблица 5.1

Результаты вычисления логических операций

X	Y	\bar{X}	\bar{Y}	$X \wedge Y$	$X \vee Y$	$X \rightarrow Y$	$X \leftrightarrow Y$
0	0	1	1	0	0	1	1
0	1	1	0	0	1	1	0
1	0	0	1	0	1	0	0
1	1	0	0	1	1	1	1

Исчисление высказываний позволяет формализовать лишь малую часть множества рассуждений, поскольку этот аппарат не позволяет учитывать внутреннюю

структуру высказывания, которая существует в естественных языках. Рассмотрим ставший классическим пример рассуждения о Сократе:

P: «Все люди смертны»

Q : «Сократ — человек»

R: «Сократ - смертен»

Используя для обозначения высказываний логические переменные P, R, Q можно составить формулу: $(P \wedge Q) \rightarrow R$, которая может быть интерпретирована как «Если все люди смертны и Сократ является человеком, то Сократ является смертным». Однако эта формула не является общезначимой, поскольку относится только к одному объекту (Сократу). Кроме того, высказывание R не выводится из P и Q, т.е., если бы мы не сформулировали R заранее, мы не смогли бы записать приведенную выше формулу.

Чтобы осуществить этот примитивный логический вывод, высказывание Q следует разделить на две части: «Сократ» (субъект) и «человек» (свойство субъекта) и представить в виде отношения «субъект — свойство», которое можно записать с помощью функции человек (Сократ).

Очевидно, что свойство конкретного субъекта с именем «Сократ» быть «человеком» может быть присуще и ряду других субъектов, что позволяет заменить константу «Сократ» на некоторую переменную, например, X. Тогда получим запись человек (X), которая обладает внутренней структурой, т.е. значение такого высказывания будет зависеть от его компонент. Записанная функция уже не является элементарным высказыванием, она называется **предикатом**.

Приведем объяснение понятия предиката, данное Д. А. Пospelовым: «Под предикатом будем понимать некоторую связь, которая задана на наборе из констант или переменных.»

Пример предиката: «P больше Q».

Если семантика P и Q не задана, то о предикате сказать особенно нечего. Пожалуй, только то, что он является антирефлексивным, антисимметричным и

транзитивным. Но при задании семантики (т.е. областей определения переменных P и Q) о предикате можно будет сказать существенно больше. Например, если P и Q — площади городов в России и Японии, то при задании списков городов и подстановке значений из этих списков в переменные мы получим отношение между двумя сущностями и сможем судить о его истинности, например:

«Площадь Волгограда больше площади Хиросимы» = T ;

«Площадь Вологды больше площади Токио» = F .

Иногда для утверждения об истинности или ложности предиката можно обойтись без подстановки. Например, если областью определения переменной X являются целые положительные числа, то предикат « $X > -5$ » будет тождественно истинен».

Основными синтаксическими единицами логики предикатов являются константы, переменные, функции, предикаты, кванторы и логические операторы. Формальный синтаксис исчисления предикатов первого порядка удобно представить в нормальной форме Бэкуса—Наура, которая традиционно применяется для записи грамматик языков программирования.

<константа> \rightarrow <идентификатор1>

<переменная> \rightarrow <идентификатор2>

<функция> \rightarrow <идентификатор3>

<предикат> \rightarrow <идентификатор4>

<терм> \rightarrow <константа> | <переменная> |

| <функция> (<список термов>)

<список термов> \rightarrow <терм> | <терм>, <список термов>

<атом> \rightarrow <предикат> | <предикат> (<список термов>)

<литера> \rightarrow <атом> | \neg <атом>

<оператор> \rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

<список переменных> \rightarrow <переменная> | <переменная> ,

<список переменных>

<квантор> \rightarrow $\langle (\exists \langle \text{список переменных} \rangle) |$

$| \langle (\forall \langle \text{список переменных} \rangle)$

<формула> \rightarrow <литера> $| \neg$ <формула> $|$ <квантор> (<формула>) $|$ (<формула>)

<оператор> (<формула>)

В данной записи любое имя в угловых скобках представляет собой тип синтаксического объекта. Определение каждого типа начинается с появления его имени в левой части каждой записи, т. е. слева от знака \rightarrow . В правой части каждой записи приводятся возможные способы организации синтаксически корректных объектов определяемого типа. Альтернативные варианты разделены знаком $|$, который можно интерпретировать как ИЛИ.

Номера идентификаторов следует трактовать в том смысле, что идентификаторы, используемые для обозначения объектов разных типов, должны быть различимыми. Например, константы обозначаются именами <идентификатор1>, которые формируются из строчных букв, причем первым символом должен быть один из следующих: a, b, c, d e, k, l, m, n, x, y, z, v, w, u.

Имена переменных <идентификатор2> должны начинаться, например, с заглавной буквы.

Идентификаторы функций <идентификатор3> состоят из строчных букв, при этом первой является f, g, h, p или q.

Имена предикатов <идентификатор4> должны состоять из прописных букв.

Функции, как и предикаты, задают некоторую связь между переменными или константами. Но отношение не характеризуются истинностным значением. С помощью функции можно представить сложный объект, например, функция fbook (Author, Tittle, Publisher, Year) представляет набор информации, характеризующей книгу. Предикат и функция отличаются также на синтаксическом уровне, а именно: функции могут являться аргументами предикатов (т.е. термами), а предикаты — нет.

Следует заметить, что в логике предикатов более высоких порядков по сравнению с первым аргументами предикатов могут быть другие предикаты. Функции с нулевым числом мест (аргументов) являются аналогами констант. Предикат без аргументов эквивалентен высказыванию.

Кванторы в логике предикатов необходимы для определения области действия переменных. Так, в логическом выводе о Сократе высказывание «Все люди смертны» можно уточнить следующим образом:

«Для всех X , если X является человеком, то X является смертным».

Введя предикаты ЧЕЛОВЕК(X) и СМЕРТЕН(X), можем составить логическую формулу ЧЕЛОВЕК(X)@СМЕРТЕН(X). Чтобы показать справедливость этой формулы для любого X , используется квантор общности:

" X — «для любого X ».

Тогда рассматриваемое утверждение запишется в виде " (X) ЧЕЛОВЕК(X)@СМЕРТЕН(X).

Кроме квантора общности в логике предикатов есть квантор существования: $\$X$ — «существует хотя бы один такой X , что ...» или «найдется хотя бы один X , такой, что ...»

Переменные, находящиеся в сфере действия кванторов, называются связанными, остальные переменные в логических формулах называются свободными. Для того чтобы можно было говорить об истинности какого-либо утверждения без подстановки значений в переменные, все входящие в него переменные должны быть связаны кванторами.

Если в логическую формулу входит несколько кванторов, необходимо учитывать их взаимное расположение. Рассмотрим возможные интерпретации логической формулы ЛЮБИТ(X , Y) с квантифицированными переменными. При этом существует несколько вариантов размещения кванторов, один из которых " $X\$Y$ ЛЮБИТ(X , Y). Эту формулу можно интерпретировать двояко:

- для любого X существует хотя бы один человек Y , которого любит X ;

*Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж*

- существует по крайней мере один человек Y , которого любят все X ;

Для устранения этой неопределенности введем скобки и порядок применения кванторов — слева направо, тогда получим следующие формулы, соответствующие интерпретациям:

$$1. (\forall X)(\exists Y) \text{ ЛЮБИТ}(X, Y).$$

$$2. (\exists Y)(\forall X) \text{ ЛЮБИТ}(X, Y).$$

Рассмотрим остальные варианты расположения кванторов и их интерпретации.

$$3. (\forall X)(\forall Y) \text{ ЛЮБИТ}(X, Y) \text{ и } (\forall Y)(\forall X) \text{ ЛЮБИТ}(X, Y) - \text{«Всеобщее человеколюбие»}.$$

4. $(\exists X)(\forall Y) \text{ ЛЮБИТ}(X, Y)$ - «Существует хотя бы один человек, который любит всех людей».

$$5. (\forall Y)(\exists X) \text{ ЛЮБИТ}(X, Y) - \text{«Каждого человека кто-нибудь любит»}.$$

6. $(\exists X)(\exists Y) \text{ ЛЮБИТ}(X, Y)$ и $(\exists Y)(\exists X) \text{ ЛЮБИТ}(X, Y)$ - «Существует хотя бы один человек, который не утратил чувства любви».

В одной логической формуле не допускается применение разных кванторов к одной переменной, например выражение $(\exists X)(\forall X)P(X)$ является недопустимым.

Отрицание кванторных выражений выполняется в соответствии со следующими правилами:

$$\neg(\forall X)P(X) = (\exists X)\neg P(X)$$

$$\neg(\exists X)P(X) = (\forall X)\neg P(X)$$

Справедливость приведенных выражений вытекает из смысла кванторов. Эти соотношения позволяют любую формулу в логике предикатов представить в виде предваренной нормальной формы (ПНФ), в которой сначала выписываются все кванторы, а затем — предикатные выражения, приведенные к виду КНФ.

Пример ПНФ: $(\forall X)(\forall Y)(\exists Z)(P(X) \rightarrow Q(Y, Z, W))$

В логике предикатов первого порядка не разрешается применение кванторов к предикатам (более высокие порядки это позволяют).

Формула, в которой все переменные связаны, называется предложением. Каждому предложению можно поставить в соответствие определенное значение — «истина» или «ложь».

Пример: пусть $f(X)$ — функция, задающая отношение «отец»; $P(X)$ — предикат, задающий отношение «человек». Тогда логическая формула $(\forall X)(P(f(X)) \rightarrow P(X))$ будет интерпретироваться как «Все существа, отцом которых является человек, - люди».

Операции в логике предикатов имеют неодинаковые приоритеты. Самый высокий приоритет имеет квантор общности, самый низкий — операция эквиваленция.

Сложные формулы в логике предикатов получаются путем комбинирования атомарных формул с помощью логических операций. Такие формулы называются правильно построенными логическими формулами (ППФ). Интерпретация ППФ возможна только с учетом конкретной области интерпретации, которая представляет собой множество всех возможных значений термов, входящих в ППФ. Для представления знаний конкретной предметной области в виде ППФ необходимо прежде всего установить область интерпретации (мир Хербранда), т.е. выбрать константы, которые определяют объекты в данной области, а также функции и предикаты, которые определяют зависимости и отношения между объектами. После этого можно построить логические формулы, описывающие закономерности данной предметной области. Записать знания с помощью логической модели не удастся в тех случаях, когда затруднен выбор указанных трех групп элементов (констант, функций и предикатов) или когда для описания этих знаний не хватает возможностей представления с помощью ППФ, например когда знания являются неполными, ненадежными, нечеткими и т.д.

Логическая модель применяется в основном в исследовательских системах, так как предъявляет очень высокие требования к качеству и полноте знаний предметной области.

Задание

Составить словарь терминов с определением понятий Логической модели представления знаний

Практическое занятие №3 Моделирование интеллектуальных систем. Предикаты

Примеры высказываний с использованием предикатов:

«Все люди смертны». Здесь предикат «смертны» относится ко всем людям.

«Некоторые животные умеют летать». Здесь предикат «умеют летать» относится к некоторым животным.

«Все квадраты являются прямоугольниками». Здесь предикат «являются прямоугольниками» относится ко всем квадратам.

Задание. Найдите предикат и к чему он относится в следующих высказываниях.

«Все люди умеют говорить».

«Некоторые животные умеют плавать».

«Каждый треугольник имеет три угла».

«Ни одна кошка не любит молоко».

«Все птицы умеют летать».

«Некоторые фрукты сладкие».

«Каждый квадрат имеет четыре стороны».

«Все рыбы живут в воде».

«Некоторые овощи полезны для здоровья».

«Каждый круг имеет бесконечное количество точек на окружности».

Практическое занятие №4 Моделирование интеллектуальных систем. Продукционная модель представления знаний

1. Продукционная модель представления знаний

Продукции (наряду с сетевыми моделями) являются наиболее популярными средствами представления знаний в информационных системах. В общем виде под продукцией понимают выражение вида $A \rightarrow B$. Обычное прочтение продукции выглядит так: ЕСЛИ A , ТО B . Импликация может истолковываться в обычном логическом смысле, как знак логического следования B из истинного A . Возможны и другие интерпретации продукции, например, A описывает некоторое условие, необходимое, чтобы можно было совершить действие B .

Продукционная модель или модель, основанная на правилах, позволяет представить знания в виде предложений типа

«Если (условие), то (действие)».

Под условием понимается некоторое предложение — образец, по которому осуществляется поиск в базе знаний, а под действием — действия, выполняемые при

Документ управляется программными средствами 1С: Колледж

Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж

успешном исходе поиска (они могут быть промежуточными, выступающими далее как условия, и терминальными или целевыми, завершающими работу системы).

При использовании продукционной модели база знаний состоит из набора правил. Программа, управляющая перебором правил, называется машиной вывода. Чаще всего вывод бывает прямой (от данных к поиску цели) или обратный (от цели для ее подтверждения – к данным). Данные — это исходные факты, на основании которых запускается машина вывода.

Если в памяти системы хранится некоторый набор продукций, то они образуют систему продукций. В системе продукций должны быть заданы специальные процедуры управления продукциями, с помощью которых происходит актуализация продукций и выполнение той или иной продукции из числа актуализированных.

В состав системы продукций входит база правил (продукций), глобальная база данных и система управления. База правил – это область памяти, которая содержит совокупность знаний в форме правил вида ЕСЛИ – ТО.

Приведем несколько примеров.

Правило 1.

ЕСЛИ (намерение — отдых) и

(дорога ухабистая)

ТО (использовать джип)

Правило 2.

ЕСЛИ (место отдыха — горы)

ТО (дорога ухабистая)

Глобальная база данных — область памяти, содержащая фактические данные (факты). Система управления формирует заключения, используя базу правил и базу данных. Существуют следующие способы формирования заключений — прямые и обратные выводы.

Правила вывода бывает удобно представлять в виде дерева решений. Граф — множество вершин, связанных дугами. Дерево — граф, не содержащий циклов.

В прямых выводах выбирается один из элементов данных, содержащихся в базе данных, и если при сопоставлении этот элемент согласуется с левой частью правила (посылкой), то из правила выводится соответствующее заключение и помещается в базу данных или исполняется действие, определяемое правилом, и соответствующим образом изменяется содержимое базы данных.

МО-09 02 07-МДК.06.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ТЕХНОЛОГИИ	С. 19/25

В обратных выводах процесс начинается от поставленной цели. Если эта цель согласуется с правой частью правила (заключением), то посылка правила принимается за подцель или гипотезу. Этот процесс повторяется до тех пор, пока не будет получено совпадение подцели с данными.

2. Примеры

Пример №1. Имеется фрагмент базы знаний из двух правил:

П1: Если (отдых – летом) и (человек – активный)

то (ехать в горы)

П2: Если (любит – солнце)

то (отдых – летом)

Предположим, в систему поступили данные — (человек – активный) и (любит – солнце).

Прямой вывод — исходя из данных, получить ответ.

1-й проход.

Шаг 1. Пробуем П1, не работает (не хватает данных (отдых–летом)).

Шаг 2. Пробуем П2, работает, в базу поступает факт (отдых–летом).

2-й проход.

Шаг 3. Пробуем П1, работает, активируется цель (ехать в горы), которая и выступает как вывод.

Обратный вывод — подтвердить выбранную цель при помощи имеющихся правил и данных.

1-й проход.

Шаг 1. Цель — (ехать в горы): пробуем П1 — данных (отдых– лето) нет, они становятся новой целью, и ищется правило, где она в правой части.

Шаг 2. Цель (отдых – летом): правило П2 подтверждает цель и активирует ее.

2-й проход.

Шаг 3. Пробуем П1, подтверждается искомая цель.

Пример №2

ЕСЛИ «двигатель не заводится» И «стартер двигателя не работает», ТО «неполадки в системе электропитания стартера»;

ЕСЛИ «животное имеет перья», ТО «животное – птица».

При большом числе продукций в продукционной модели усложняется проверка непротиворечивости системы продукций, то есть множества правил. Поэтому число продукций, с которыми работают современные системы искусственного интеллекта, как правило, не превышает тысячи.

Продукционная модель привлекает разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического вывода.

Приведем сильные и слабые стороны систем продукций.

Сильные стороны систем продукций:

- модульность;
- единообразие структуры (основные компоненты продукционной системы могут применяться для построения интеллектуальных систем с различной проблемной ориентацией);
- естественность (вывод заключения в продукционной системе во многом аналогичен процессу рассуждения эксперта);
- гибкость родовидовой иерархии понятий, которая поддерживается только как связь между правилами (изменение правила ведет за собой изменение в иерархии);
- простота создания и понимания отдельных правил;
- простота пополнения и модификации;
- простота механизма логического вывода.

Слабые стороны систем продукций

1. процесс вывода менее эффективен, чем в других системах, поскольку большая часть времени при выводе затрачивается на непроизводительную проверку применимости правил;
2. сложно представить родовидовую иерархию понятий;
3. неясность взаимных отношений правил;
4. сложность оценки целостного образа знаний;
5. отличие от человеческой структуры знаний;
6. отсутствие гибкости в логическом выводе

Представление знаний с помощью продукций иногда называют «плоским», так как в продукционных системах отсутствуют средства для установления иерархий правил. Объем знаний продукционных систем растет линейно, по мере включения в нее новых фрагментов знаний, в то время как в традиционных алгоритмических системах, использующих деревья решений, зависимость между объемом база знаний и количеством знаний является логарифмической.

МО-09 02 07-МДК.06.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ТЕХНОЛОГИИ	С. 21/25

Имеется большое число программных средств, реализующих продукционный подход: OPS5, EXSYS RuleBook, ЭКСПЕРТ, ЭКО, G2 и др.

Модификации продукционной модели

Продукционная модель часто дополняется определённым порядком, вводимым на множестве продукций, что упрощает механизм логического вывода. Порядок может выражаться в том, что отдельная следующая по порядку продукция может применяться только после попыток применения предшествующих ей продукций. Примерно похожее влияние на продукционную модель может оказать использование приоритетов продукций, означающее, что в первую очередь должна применяться продукция, имеющая наивысший приоритет.

Рост противоречивости продукционной модели может быть ограничен путём введения механизмов исключений и возвратов. Механизм исключений означает, что вводятся специальные правила-исключения. Их отличает большая конкретность в сравнении с обобщёнными правилами. При наличии исключения основное правило не применяется. Механизм возвратов же означает, что логический вывод может продолжаться в том случае, если на каком-то этапе вывод привёл к противоречию. Просто необходимо отказаться от одного из принятых ранее утверждений и осуществить возврат к предыдущему состоянию.

Противоречия в базах знаний на языке Пролог выявляются автоматически за счет использования автоматического доказательства теорем со встроенным в систему Пролог механизмами перебора с возвратами, организующего поиск информации в базах знаний и выводом найденной информации в качестве результатов информационного поиска.

Антецеденты и консеквенты правил формируются из атрибутов и значений, например:

Любое правило состоит из одной (или нескольких) пары атрибут-значение. В рабочей памяти продукционной системы хранятся пары атрибут-значение, истинность которых установлена в процессе решения конкретной задачи к некоторому текущему моменту времени. Содержимое рабочей памяти изменяется в процессе решения задачи. Это происходит по мере срабатывания правил. Правило срабатывает, если при сопоставлении фактов, содержащихся в рабочей памяти, с антецедентом анализируемого правила имеет место совпадение, при этом заключение сработавшего правила заносится в рабочую память. Поэтому в процессе логического вывода объём фактов в рабочей памяти, как правило, увеличивается (уменьшается он может в том случае, если действие какого-нибудь правила состоит в удалении фактов из рабочей памяти). В процессе логического вывода каждое правило из базы правил может сработать только один раз.

При описании реальных знаний конкретной предметной области может оказаться недостаточным представление фактов с помощью пар атрибут-значение. В этом случае отдельная сущность предметной области рассматривается как объект, а данные, хранящиеся в рабочей памяти, показывают значения, которые принимают атрибуты этого объекта.

МО-09 02 07-МДК.06.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ТЕХНОЛОГИИ	С. 22/25

Примеры триплетов

Собака – кличка – Граф;

Собака – порода – ризеншнауцер;

Собака – окрас – чёрный.

Одним из преимуществ такого представления знаний является уточнение контекста, в котором применяются правила. Например, правило, относящееся к объекту «собака», должно быть применимо для собак с любыми кличками, всех пород и окрасок. С введением триплетов правила из базы правил могут срабатывать более одного раза в процессе одного логического вывода, поскольку одно правило может применяться к различным экземплярам объекта (но не более одного раза к каждому экземпляру).

Существуют два типа продукционных систем – с прямыми и обратными выводами. Прямые выводы реализуют стратегию «от фактов к заключениям». При обратных выводах выдвигаются гипотезы вероятностных заключений, которые могут быть подтверждены или опровергнуты на основании фактов, поступающих в рабочую память. Существуют также системы с двунаправленными выводами.

Основные достоинства продукционных систем связаны с простотой представления знаний и организации логического вывода. К недостаткам систем продукций можно отнести следующее:

- Отличие от структур знаний, свойственных человеку;
- Неясность взаимных отношений правил;
- Сложность оценки целостного образа знаний;
- Низкая эффективность обработки знаний.

При разработке небольших систем (десятки правил) проявляются в основном положительные стороны систем продукций, однако при увеличении объёма знаний более заметными становятся слабые стороны.

Задание.

Составить 15 вопросов с ответами по материалам лекции

Практическое занятие №5 Моделирование интеллектуальных систем. Семантические ИС

Примеры семантических информационных сетей и систем

Группа учёных из университета Пенсильвании (Pennsylvania State University) объявила о регистрации патента на «семантическую» систему визуального поиска ALIPR (Automatic Linguistic Indexing of Pictures in Real- Time). Суть нового сервиса в объединении «пиксельного» и смыслового анализа изображений. В базе ALIPR

*Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж*

находится более миллиона изображений, к каждому из которых проставлены теги – группа смысловых «ключей». Основная идея в том, что восприятие тех или иных объектов у пользователей Интернета всё же подчиняется законам нормального распределения, и, следовательно, наиболее популярные связки «понятие – объект» должны совпасть с ожиданиями. В настоящий момент система оперирует 332 понятиями, но это не означает, что найти можно картинки лишь по столь ограниченному набору слов.

WordNet — это семантическая сеть для английского языка, разработанная в Принстонском университете, и выпущенная вместе с сопутствующим программным обеспечением под некомпилетной свободной лицензией.

Freebase – семантическая база данных. Freebase работает не только через текстовый поиск, а что наиболее важно, и через — MQL (Metaweb Query Language). MQL схоже с JSON, текстовый формат обмена данными. База знаний пользовательского интерфейса включает в себя следующую информацию:

- описание команд пользовательского интерфейса – описываются классы команд пользовательского интерфейса, которые служат для инициирования пользователем каких-либо действий в системе.
- описание структуры главного меню;
- описание внешних языков представления знаний;
- описание используемых компонентов – используются интерфейсом для организации диалога с пользователем. К примеру, при установке содержимого sc-рамки пользователю отображается список редакторов, которые имеются в базе знаний;
- протокол пользовательских действий – строится во время работы системы. Он необходим для построения портрета пользователя, а также обеспечения отмены пользовательских действий;
- портрет пользователя – строится на основании анализа действий пользователя. Включает в себя всю необходимую информацию о пользователе. В качестве такой информации может выступать предпочитаемая стилистика размещения, наиболее частоиспользуемые внешние языки; акт обмена данными), только с более широкими возможностями. С его помощью можно составить любой запрос в Freebase и ответом будет тот же запрос, но уже со вставленными результатами поиска.

Семантическая технология проектирования баз знаний, программ, машин обработки знаний, пользовательских интерфейсов, интеллектуальных справочных систем, интеллектуальных обучающих систем

Машина обработки знаний пользовательского интерфейса имеет свою специфику – кроме обычных операций обработки знаний в ней присутствуют

“эффекторные” и “рецепторные” операции. “Эффекторные” операции осуществляют вывод информации пользователю, реагируя на событие в памяти системы. “Рецепторные” операции реагируют на события пользователя (нажатие клавиши, перемещение мыши) и изменяют состояние памяти системы. В машину обработки знаний пользовательского интерфейса обязательно входят следующие операции:

1. Операции работы с командами:
 - добавление дочерней команды в неатомарную команду;
 - удаление дочерней команды из неатомарной команды;
2. Операции работы с компонентами:
 - добавление компонента;
 - удаление компонента;
3. Операции работы с окнами:
 - создание окна указанного типа;
 - удаление окна;
 - открытие/закрытие окна;
4. “Эффекторные” операции:
 - инициирование вывода ответа на вопрос, заданный пользователем;
 - вывод информационной конструкции пользователю;
5. “Рецепторные” операции (список может быть расширен в зависимости от используемых внешних устройств):
 - нажатие кнопки на клавиатуре;
 - нажатие клавиши мыши;
 - вращение колеса прокрутки мыши;

В состав предлагаемой семантической технологии компонентного проектирования пользовательских интерфейсов интеллектуальных систем входит:

- унифицированная семантическая модель пользовательских интерфейсов интеллектуальных систем;
- библиотека совместимых ip-компонентов пользовательских интерфейсов;
- инструментальные средства компонентного проектирования пользовательских интерфейсов;
- методика компонентного проектирования пользовательских интерфейсов;
- методика обучения проектированию пользовательских интерфейсов;
- интеллектуальная help-система по семантической технологии компонентного проектирования пользовательских интерфейсов интеллектуальных систем.

Поставленные задачи соответствуют разработке выше перечисленных подсистем, которые входят в состав технологии.

Интеллектуальная help-система, по семантической технологии проектирования пользовательских интерфейсов интеллектуальных систем, решает задачу помощи разработчикам в процессе проектирования пользовательских интерфейсов. Данная интеллектуальная система строится с использованием семантической технологии проектирования интеллектуальных систем. Её база знаний содержит следующие разделы:

1 Унифицированная семантическая модель пользовательских интерфейсов интеллектуальных систем. В данном разделе содержится полное формальное описание модели пользовательских интерфейсов, это дает возможность пользователю (разработчику) значительно быстрее усвоить теорию и понять принципы, лежащие в основе проектируемых им интерфейсов.

2 Библиотека совместимых ip-компонентов пользовательских интерфейсов интеллектуальных систем. В этом разделе приводится полное формальное описание библиотеки ip-компонентов. Это позволяет разработчику значительно быстрее освоить принципы работы с библиотекой и тем самым сократить сроки проектирования.

3 Интегрированные средства проектирования пользовательских интерфейсов. Раздел базы знаний, посвященный писанию средств, которые позволяют автоматизировать и интеллектуализировать процесс проектирования.

4 Методика проектирования пользовательских интерфейсов интеллектуальных систем. Содержит детальное описание процесса проектирования.

5 Методика проектирования пользовательских интерфейсов интеллектуальных систем. Содержит детальное описание процесса проектирования.

6 Методика обучения проектированию пользовательских интерфейсов интеллектуальных систем. Данный раздел базы знаний полностью посвящен процессу обучения проектированию. Он содержит информацию, которая должна помочь пользователю в освоении самой технологии.

7 Help-система по семантической технологии проектирования пользовательских интерфейсов интеллектуальных систем. Раздел, который призван помочь пользователю освоить саму help-систему.

Включение интеллектуальной help-системы в состав технологии позволяет не только сократить сроки проектирования, но и снизить требования, предъявляемые к начальной квалификации разработчика.

Задание

Создайте интеллект-карту по материалам лекции **Примеры семантических информационных сетей и систем**