

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)
Кафедра информационных систем и технологий

УТВЕРЖДАЮ
Проректор по учебной работе
Ю.Г. Локтионова
« 06 » июня 2019 г.



Представление знаний в информационных системах:
методические указания к практическим занятиям для бакалавров
направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 16.06.2023 12:33:44
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

Курс 2019

УДК 004.832.3

Составитель: Ю.А. Халин

Рецензент

Кандидат технических наук, доцент С.Ю. Сазонов

Представление знаний в информационных системах: методические указания к практическим занятиям / Юго-Зап. гос. ун-т; сост. Ю.А. Халин. Курск, 2019. 28 с. Библиогр.: с. 28.

Описываются модели представления знаний, основанные на положения логики высказываний. Изложены краткие теоретические сведения из математической логики, приведены примеры решения задач, а также задачи для самостоятельного решения.

Методические рекомендации предназначены для студентов, обучающихся по направлениям 02.03.03 Математическое обеспечение и администрирование информационных систем.

Текст печатается в авторской редакции.

Подписано в печать *06.08.19* . Формат 60x84 1/16.

Усл.печ. л. 1,59 п.л . Уч.-изд. л. 1,28 . Тираж 100 экз. Заказ. *511* Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

Практическая работа № 1 Логические модели представления знаний	4
Практическая работа №2. Семантические сети. Фреймы	9
Практическая работа №3. Продукционные правила	16
Практическая работа №4. Представление нечётких и неточных знаний.....	19
Практическая работа №5 Методика построения и архитектура экспертных систем	22

Практическая работа № 1 Логические модели представления знаний

Основные положения логики высказываний.

Высказывание - повествовательное предложение (*утверждение, суждение*), о котором имеет смысл говорить, что оно *истинно (значение 1)* или *ложно (значение 0)*.

Табличное задание *логических операций*.

Исходные высказывания		Вид операции							
A	B		$A \wedge B$	$A \vee B$	A	$A \leftrightarrow B$	$A B$	$A \uparrow B$	
0	0	1	0	0	1	1	1	1	0
0	1	1	0	1	1	0	1	0	1
1	0	0	0	1	0	0	1	0	1
1	1	0	1	1	1	1	0	0	0

Формулой алгебры высказываний называется выражение, составленное из переменных высказываний с помощью операций над высказываниями и обращающееся в конкретное высказывание при подстановке вместо переменных высказываний конкретных высказываний.

Подформулой формулы **A** называется любое подслово (часть) слова **A**, которое само являющееся формулой.

Формула алгебры высказываний называется *тавтологией* или *тождественно истинной*, если при любых значениях входящих в нее переменных высказываний она превращается всегда в истинное высказывание.

Формула алгебры высказываний называется *тождественно ложной* или *противоречием*, если она ложна, как бы ни определялись значения переменных высказываний, входящих в эту формулу

Формула алгебры высказываний называется *выполнимой* (разрешимой), если при определенном наборе значений переменных высказываний она превращается в истинное высказывание.

Формулы **A** и **B** называются эквивалентными (равносильными). (обозначается $A \equiv B$), если при любых значениях переменных высказываний значение **A** совпадает со значением **B**.

Если **V** является подформулой формулы **F** и **V** эквивалентна **W**, то формула **F₁**, полученная из **F** заменой подформулы **V** на **W**, эквивалентна формуле **F**.

Если две формулы эквивалентны $A \equiv B$, то формула $A \leftrightarrow B$ тождественно истинна.

Практические задания по вариантам

Задача 1. Определите, является ли данное выражение формулой. Если это формула, то выпишите последовательность построения формулы.

Вариант	Выражение
1	$(A_0 \& A_1) A_2 \neg A_3$
2	$(A_0 \& A_1) \Rightarrow A_5$
3	$((A_3 \Rightarrow A_0) \& \neg A_0)$
4	$((\neg A_0 \Rightarrow A_1) \Rightarrow \neg(A_2 \vee A_3))$
5	$((X \Rightarrow Y) \& Z)$
6	$(X \& Y) \vee Z$
7	$(\neg A \Rightarrow B \& C) \vee (D \& \neg A \Rightarrow C)$
8	$((A_0 \Rightarrow A_1) \Rightarrow ((A_0 \neg A_1) \Rightarrow \neg A_1))$
9	$A_0 \Rightarrow (\neg A_1 \vee (A_1 \& A_2))$
10	$A_1 \Rightarrow A_2 \Rightarrow A_3 \Rightarrow \neg A_1 \Rightarrow \neg A_2$

Задача 2. Сколькими способами можно расставить скобки в последовательности, чтобы получилась формула. Выписать все возможные получаемые формулы.

Вариант	Выражение
1	$A_1 \Rightarrow A_2 \Rightarrow A_3 \Rightarrow \neg A_1 \Rightarrow \neg A_2$
2	$A_0 \Rightarrow \neg A_1 \vee A_1 \& A_2$
3	$\neg A_0 \Rightarrow A_1 \Rightarrow \neg A_2 \vee A_3$
4	$X \Rightarrow Y \& Z$
5	$A_0 \Rightarrow A_1 \Rightarrow A_0 \Rightarrow \neg A_1$
6	$A_0 \& A_1 \Rightarrow A_2 \& \neg A_3$
7	$A_0 \& A_1 \Rightarrow A_5$
8	$A_3 \Rightarrow A_0 \& \neg A_0$
9	$\neg A \Rightarrow B \& C \vee D \& \neg A \Rightarrow C$
10	$X \& Y \vee Z$

Задача 3. Выписать все подформулы данной формулы.

Вариант	Выражение
1	$((A_0 \Rightarrow A_1) \& (A_1 \Rightarrow A_2)) \Rightarrow (\neg A_0 \vee A_2)$

2	$(\neg A_0 \Rightarrow A_1) \Rightarrow (\neg A_2 \vee A_3)$
3	$((A_0 \Rightarrow \neg A_1) \vee A_1) \& A_2$
4	$((A_0 \Rightarrow A_1) \Rightarrow ((A_0 \Rightarrow \neg A_1) \Rightarrow \neg A_1))$
5	$(\neg A \Rightarrow B \wedge C) \vee ((D \wedge \neg A) \Rightarrow \neg C)$
6	$(\neg(A \Rightarrow B) \& C) \vee (((D \& (\neg A)) \Rightarrow C)$
7	$((A \wedge B) \Rightarrow C) \wedge (D \vee (A \Leftrightarrow C))$
8	$(\neg A \Rightarrow B \wedge C) \vee (D \wedge \neg A \Rightarrow \neg C)$
9	$(A_0 \Rightarrow (A_1 \Rightarrow A_0)) \Rightarrow (\neg A_1)$
10	$((\neg A_0 \Rightarrow A_1) \Rightarrow \neg(A_2 \vee A_3))$

Задача 4. Указать тип формулы. Доказать вывод.

Вариант	Выражение
1	$(A \wedge B \Rightarrow C) \wedge (D \vee A \Leftrightarrow C)$
2	$(\neg(A \Rightarrow B) \& C) \vee (((D \& (\neg A)) \Rightarrow C)$
3	$(A \Rightarrow B) \wedge (B \Rightarrow A)$
4	$(A \vee B) \wedge (B \Rightarrow A)$
5	$(A \wedge B \Rightarrow C) \wedge (D \wedge A \Rightarrow C)$
6	$(\neg A \Rightarrow B) \wedge (B \Rightarrow A)$
7	$((A \wedge B) \Rightarrow C) \wedge (D \vee (A \Leftrightarrow C))$
8	$(\neg A \Rightarrow \neg B) \wedge (B \Rightarrow A)$
9	$(\neg A \Rightarrow B \wedge C) \vee ((D \wedge \neg A) \Rightarrow \neg C)$
10	$(\neg A \Rightarrow B \wedge C) \vee (D \wedge \neg A \Rightarrow C)$

Задача 5. С помощью таблиц истинности, а так же с помощью эквивалентных преобразований проверить на эквивалентность формулы.

Вариант	Формулы	
1	$(A \Rightarrow A \wedge C) \vee (B \wedge \neg A)$	$(A \Rightarrow (A \wedge C \Leftrightarrow B))$
2	$(A \Rightarrow A \wedge C) \vee (B \wedge \neg A)$	$(A \Rightarrow (C \Rightarrow B))$
3	$(A \Rightarrow A \wedge C) \vee (B \wedge \neg A)$	$(A \wedge C) \vee (\neg B \vee \neg A \wedge B)$
4	$(A \Rightarrow A \wedge C) \vee (B \wedge \neg A)$	$\neg A \vee B \vee \neg C$
5	$(A \wedge B \Rightarrow \neg B \vee A \wedge C)$	$(\neg B \Leftrightarrow A) \Rightarrow C$
6	$(A \wedge B \Rightarrow \neg B \vee A \wedge C)$	$(A \mid B) \vee C$
7	$(A \wedge B \Rightarrow \neg B \vee A \wedge C)$	$(A \uparrow C) \mid B$
8	$(A \wedge B \Rightarrow \neg B \vee A \wedge C)$	$B \Rightarrow A \wedge C \vee \neg A \wedge B$
9	$B \Rightarrow A \wedge C \vee \neg A \wedge B$	$(A \Rightarrow (A \wedge C \Leftrightarrow B))$
10	$B \Rightarrow A \wedge C \vee \neg A \wedge B$	$(A \wedge C) \vee (\neg B \vee \neg A \wedge B)$

Задача 6. Представьте логическими формулами пословицы и поговорки.

Вариант	Выражение
1	Без еды не будет и беседы.
2	Без недостатка только Бог, без грязи только вода.
3	Близкому не говори ложь, постороннему не говори правду.
4	Если тебе угощать нечем – хоть говори ласково.
5	Когда грома много – дождя мало.
6	Гнев впереди, ум позади.
7	Доброе слово — половина счастья.
8	Если уважаешь отца, люби и сына; если уважаешь хозяина, корми и его собаку.
9	Кочерга длинная – не обожжешь руки; много родных – люди не обидят.
10	Кто много видит – становится умнее, кто много говорит – становится красноречивее.

Задача 7. Доказать законы логики.

Вариант	Название законов.
1	Закон двойного отрицания. Идемпотентность дизъюнкции.
2	Коммутативный закон конъюнкции. Закон тождества.
3	Коммутативный закон дизъюнкции. Идемпотентность конъюнкции.
4	Ассоциативность конъюнкции. Закон поглощения для дизъюнкции.
5	Ассоциативность дизъюнкции. Закон поглощения для конъюнкции.
6	Дистрибутивность конъюнкции относительно дизъюнкции. Закон противоречия.
7	Дистрибутивность дизъюнкции относительно конъюнкции. Свойства единицы.
8	Закон де Моргана - отрицание конъюнкции. Выражение импликации через дизъюнкцию.
9	Закон де Моргана - отрицание дизъюнкции. Выражение импликации через конъюнкцию.
10	Закон исключения третьего. Свойства нуля.

Задача 8. При каких значениях переменных формула ложна.

Вариант	Формула
1	$((X \Rightarrow (Y \wedge Z)) \Rightarrow (\neg Y \Rightarrow \neg X)) \Rightarrow \neg Y$

2	$((X \vee Y) \vee Z) \Rightarrow ((X \vee Y) \wedge (X \vee Z))$
3	$((X \vee Y) \wedge ((Y \vee Z) \wedge (Z \vee X))) \Rightarrow ((X \wedge Y) \wedge Z)$
4	$((X \vee Y) \Rightarrow ((\neg X \wedge Y) \vee (X \wedge \neg Y)))$
5	$((X \Rightarrow Y) \Rightarrow (Y \Rightarrow X))$
6	$((Q \Rightarrow (P \wedge R)) \wedge \neg((P \vee R) \Rightarrow Q))$
7	$\neg(X \Rightarrow \neg X)$
8	$((X \vee Y) \wedge Z) \Rightarrow ((X \vee Y) \wedge (X \vee Z))$
9	$((X \Rightarrow (Y \wedge Z)) \Rightarrow (\neg Y \vee \neg X)) \Rightarrow \neg Y$
10	$((X \vee Y) \Rightarrow (Y \Rightarrow X))$

Практическая работа №2. Семантические сети. Фреймы

Краткие теоретические сведения.

Семантическая сеть – это один из способов представления знаний. Изначально семантическая сеть была задумана как модель представления долговременной памяти в психологии, но впоследствии стала одним из способов представления знаний в ЭС.

Семантика – означает общие отношения между символами и объектами из этих символов.

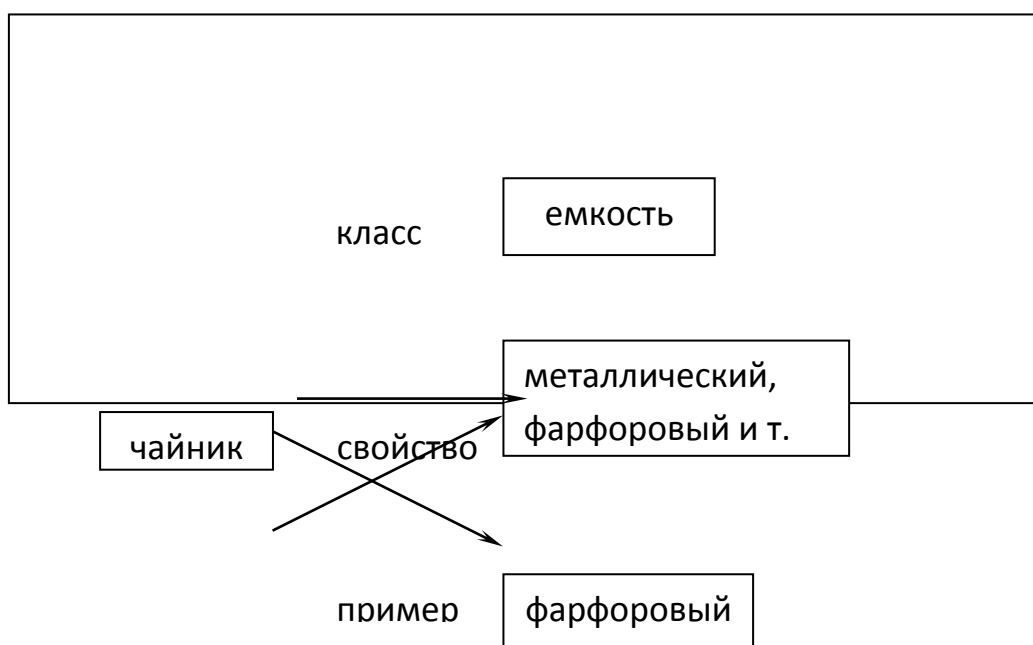


Рис.1. Простейший образец семантической сети.

Вершины – это объекты, дуги – это отношения. Семантическая модель не раскрывает сама по себе каким образом осуществляется представление знаний. Поэтому семантическая сеть рассматривается как метод представления знаний и структурирования знаний. При расширении семантической сети в ней возникают другие отношения:

IS – A (принадлежит) и PART OF (является частью) отношение:

целое → часть.

Ласточка IS – A птица, «нос» PART OF «тело». Например:

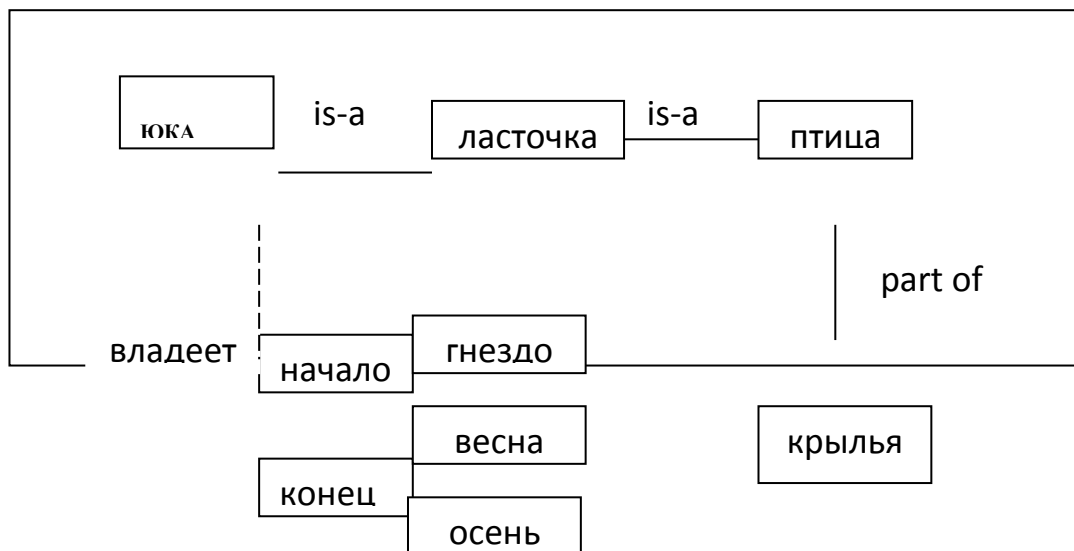


Рис.2. Расширение семантической сети

Могут быть и другие отношения: владеет. Тогда семантическая сеть расширяется иерархически (вершина имеет две ветви). Кроме того, можно расширить сеть и другим отношением:

период → «весна – лето».

Получается иерархическая структура понятия ЮКО. Можно разбить на подсхемы. Большой проблемой для семантических сетей является то, что результат вывода не гарантирует достоверности, так как вывод есть просто наследование свойств ветви is-a.

Для отображения иерархических отношений между объектами и введения единой семантики в семантические сети было предложено использовать процедурные сети. Сеть строится на основе класса (понятия); вершины, дуги и процедуры представлены как объекты.

1. Порядок выполнения работы

1. Изучить теоретическую часть по приведенным выше данным и дополнительной литературе;
2. Просмотреть демонстрационный пример;
3. Получить у преподавателя вариант задания для выполнения;

4. Построить семантическую модель заданного объекта;
5. Реализовать программу с использованием семантической модели

2. Варианты заданий

Используя соответствующие дуги построить семантическую сеть, касающуюся:

1. географии какого-либо региона. Дуги: государство, страна, континент, широта.
2. диагностики глазных заболеваний. Дуги: категории болезней, патофизиологическое состояние, наблюдения, симптомы.
3. распознавания химических структур. Дуги: формула вещества, свойства вещества, область применения, меры предосторожности.
4. процедуры поиска полезных ископаемых. Дуги: наименование ископаемого, расположение месторождения, глубина залегания, методы добычи.
5. судебной процедуры. Дуги: юридическое лицо, событие, меры воздействия, способы расследования.
6. распределения продуктов по магазинам. Дуги: источник снабжения, наименование продукта, способ транспортировки, конечный пункт транспортировки.
7. определения принадлежности животного к определенному виду, типу, семейству. Дуги: место обитания, строение, особенности поведения, вид питания.
8. классификации пищевых продуктов. Дуги: наименование продукта, составляющие части, способ приготовления, срок хранения.
9. распознавания типа компьютера. Дуги: страна изготовитель, стандартная конфигурация, область применения, используемое программное обеспечение.
10. иерархической структуры БД. Дуги: система, состояние, назначение, взаимодействие составляющих.

4. Контрольные вопросы

1. Что такое семантическая сеть и для чего ее применяют?
2. В чем состоит идея создания семантической сети?
3. Каким образом представляются данные в семантической сети?
4. Существуют ли ограничения на число связей элементов, свойств и сложность при построении семантической сети?

5. Какие отношения предложены в качестве операторов отношения для группировки вершин?

Использование фреймов для представления знаний

Цель работы: Научиться использовать фреймы для представления знаний в интеллектуальных системах

1. Теоретическая часть

Фреймы - один из распространенных формализмов представления знаний в ЭС. Фрейм можно представить себе как структуру, состоящую из набора ячеек - слотов. Каждый слот состоит из имени и ассоциируемых с ним значений. Значения могут представлять собой данные, процедуры, ссылки на другие фреймы или быть пустыми. Такое построение оказывается очень удобным для моделирования аналогий, описания областей с родовидовыми связями понятий и т.п.

Любой фрейм состоит из некоторых составляющих, имена и содержание которых описано ниже:

1. **Имя фрейма.** Это идентификатор, присваиваемый фрейму, фрейм должен иметь имя уникальное в данной фреймовой системе.
2. **Имя слота.** Это идентификатор, присваиваемый слоту; слот должен иметь уникальное имя во фрейме, к которому он принадлежит. Обычно имя слота не несет никакой смысловой нагрузки и является лишь идентификатором данного слота.
3. **Указатели наследования.** Эти указатели касаются только фреймовых систем иерархического типа, основанные на отношениях “абстрактное-конкретное”, они показывают, какую информацию об атрибутах слотов во фрейме верхнего уровня наследуют слоты с такими же именами во фрейме нижнего уровня. Типичные указатели наследования Unique (U: - уникальный), Same (S: такой же), Range (R: установление границ), Override (O: игнорировать) и т.п. U показывает, что фрейм может иметь слоты с разными значениями: S - все слоты

должны иметь одинаковые значения, R - значение слотов фрейма нижнего уровня должны находиться в пределах, указанных значениями слотов фрейма верхнего уровня, O - при отсутствии указания значение слота фрейма верхнего уровня становится значением слота фрейма нижнего уровня, но в случае определения нового значения слотов фреймов нижних уровней указываются в качестве значений слотов.

4. Указание типа данных. указывается, что слот имеет численное значение, либо служит указателем другого фрейма. К типам данных относятся:

FRAME (указатель), INTEGER (целый), REAL (действительный), BOOL (булев), LISP (присоединенная процедура), TEXT (текст), LIST (список), TABLE (таблица), EXPRESSION (выражение) и др.

5. Значение слота. Пункт ввода значения слота. Значение слота должно совпадать с указанным типом данных этого слота, кроме того должно выполняться условие наследования.
6. Демон. Здесь дается определение демонов типа IF-NEEDED, IF-ADDED, IF-REMOVED и т.д. Демоном называется процедура, автоматически запускаемая при выполнении некоторого условия. демоны запускаются при обращении к соответствующему слоту. Кроме того, демон является разновидностью присоединенной процедуры.
7. Присоединенная процедура. В качестве значения слота можно использовать программу процедурного типа. Когда мы говорим, что в моделях представления знаний фреймами объединяются процедурные и декларативные знания, то считаем демоны и присоединенные процедуры процедурными знаниями.

Особенностью иерархической структуры является то, что информация об атрибутах фрейма на верхнем уровне совместно используется всеми фреймами нижних уровней, связанных с ним.

Например: Фреймовое представление конференции.

Иерархические фреймовые структуры базируются на отношениях IS – A между фреймами, описывающими некоторую конференцию. Все фреймы должны содержать информацию о ДАТЕ, МЕСТЕ, НАЗВАНИИ ТЕМЫ,

ДОКЛАДЧИКЕ. Таким образом, на самом верхнем уровне определен фрейм КОНФЕРЕНЦИЯ.

Конференции разделяются на коммерческие и по развитию. Они составляют дочерние фреймы. В них могут быть добавлены слоты: объем торговли и бюджет.

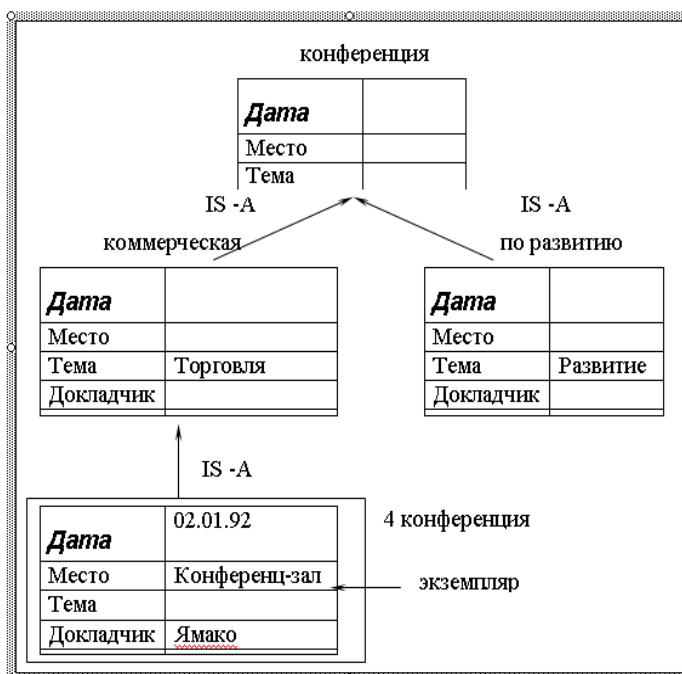


Рис.3. Пример фреймовой модели

2. Порядок выполнения работы:

1. Изучить теоретическую часть по приведенным выше данным и дополнительной литературе.
2. Просмотреть демонстрационный пример.
3. Получить у преподавателя вариант задания для выполнения.
4. Построить фреймовую модель заданного объекта;
5. Реализовать программу с использованием фреймовой модели

3. Варианты заданий

Используя фреймовую модель представления знаний реализовать структуру отношений, описывающие следующие ситуации:

1. экзамен по дисциплине за семестр у преподавателя при составляющих: семестр, экзамен, преподаватель, оценка, студент, получать.
2. ведомость при составляющих: дисциплина, студент, экзамен, семестр, преподаватель, оценка.
3. конференция по коммерческим вопросам при составляющих: дата, место проведения, тема, цель выступающие.
4. получение оценки при составляющих: преподаватель, студент, оценка, получать.
5. использования изделия при составляющих: организация, разработка технологического решения, исследование «физического эффекта», методы создания изделия.
6. информационная структура БД в машиностроении при составляющих: физические эффекты, технические решения, изделия, объект поставки изделия, приборы и стенды, нормативы.
7. классификация продукта при составляющих: название, область применения, способ хранения, способ транспортировки.
8. аудитория (описание) при составляющих: вместимость, назначение, составляющие, местонахождение.
9. животный мир при составляющих: вид, тип, среда обитания, особенности поведения.

4. Контрольные вопросы

1. Что представляет из себя фрейм, его составные части?
2. Что такое слот и из каких частей он состоит?
3. Для чего служат имя фрейма и имя слота?
4. Для чего служат указатели наследования?
5. для чего служат указание типа данных, демон?
6. Для чего служат присоединенная процедура и значение слота?

Практическая работа №3. Продукционные правила

Представление знаний с помощью правил продукции – самая распространенная форма реализации БЗ. С помощью продукции можно описать практически любую систему знаний.

Правила продукций представлены в виде импликации:

$$p_i : s_i \rightarrow d_i,$$

где p_i - правило продукции,

s_i - условие применения правила,

d_i - результат применения правила.

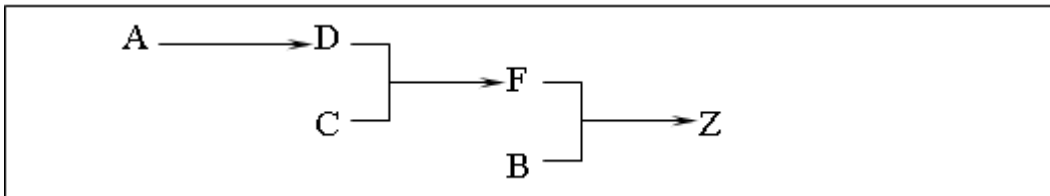


Рис.5. Пример использования правил продукции:

1. Если есть цены на выпускаемые изделия (A) - завод отпускает продукцию (D).
2. Если завод выпускает продукцию и выполняет план по ее реализации (C) - рабочие получают премию (F).
3. Если рабочие получают премию и растет производительность производства (B)- завод производит продукцию сверх плана (Z).

Рассмотрим цепочки выводов.

Прямой способ рассуждения.

По известным фактам отыскивается заключение, которое следует из этих фактов и накапливается рабочая память.

Это приводит к выполнению 2 правила.

$C \& D \rightarrow F$, и факт «F» помещается в рабочую память. Тогда опять проверяются правила из базы. Первое правило выполняется $F \& B \rightarrow Z$, вследствие этого Z заносится в рабочую память. А так как Z является целью, то поиск заканчивается. Этот метод называется прямой цепочкой рассуждений, поскольку поиск новой информации происходит в направлении стрелок, разделяющих левые и правые части правил.

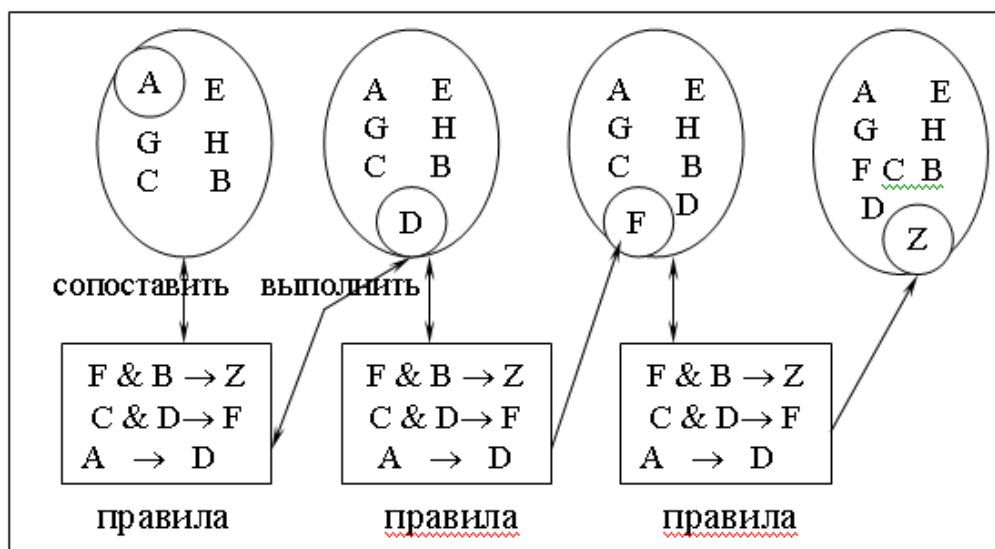


Рис.6. Пример реализации прямой цепочки рассуждений

Обобщённый алгоритм работы системы, реализующий прямую цепочку рассуждений, можно свести к следующему :

1. Определить исходное состояние.
2. Занести переменную условия в очередь переменных логического вывода, а её значение - в список переменных.
3. Просмотреть список переменных и найти ту переменную, имя которой стоит в начале очереди переменных логического вывода. Если переменная найдена, записать в указатель переменных условия номер правила и число 1. Если переменная не найдена, перейти к шагу 6.
4. Присвоить значения не проинициализированным переменным условной части найденного правила (если такие есть). Имена переменных содержатся в списке переменных условия. Проверить все условия правила и в случае их истинности обратиться к части ТО правила.
5. Присвоить значение переменной, входящей в часть ТО правила, и поместить её в конец очереди переменных логического вывода.
6. Удалить переменную, стоящую в начале очереди переменных логического вывода, если она больше не встречается в условной части какого-либо правила.

Закончить процесс рассуждений, как только опустеет очередь переменных логического вывода. Если же в очереди ещё есть переменные, вернуться к шагу 3.

1. Порядок выполнения работы:

1. Изучить теоретическую часть по приведенным выше данным и дополнительной литературе.

2. Просмотреть демонстрационный пример.
3. Получить у преподавателя вариант задания для выполнения.
4. Построить прямую цепочку рассуждений
5. Реализовать программу для прямой цепочки рассуждений

3. Варианты заданий

Реализовать прямую цепочку рассуждений для следующих задач:

1. прогнозирование неисправностей электронной аппаратуры
2. прогнозирование неисправностей автомобиля
3. прогнозирование заболеваний (по выбору)
4. прогнозирование (по выбору)
 - а. спортивных мероприятий
 - б. телепередач
 - с. природных катаклизмови т.п.
5. классификация объектов (по выбору)
- б. задачи информационно-советующего характера (по выбору)
 - а. помощник заведующего склада
 - б. помощник аптекаря
 - с. помощник оператора справочной службы
 - д. выбор должности
 - е. проведение отпускаи т.п.

4. Контрольные вопросы

1. Что такое правила продукции и в чем их сущность?
2. В чем отличие прямой цепочки рассуждений от обратной цепочки рассуждений?
3. Из каких частей состоит продукционная система?
4. Значение и применение частей продукционной системы для представления знаний?

Практическая работа №4. Представление нечётких и неточных знаний

Не всегда можно описать событие с помощью точно определенных правил. Люди не всегда могут ответить на вопросы точно. Можно ли узнать, какая у человека температура, если он говорит, что слегка заболел? Скорее всего нет! Такие слова, как «высокий», «горячий» и «легкий», «растет» или «падает», представляют собой *лингвистические переменные*, которые нельзя определить одним значением. Использование этих понятий при формировании правил называется *нечеткой логикой*.

Понятие «падает» - также *лингвистическая переменная*, используемая в правилах, описывающих фондовую биржу. Применяя *лингвистические переменные* можно вычислить значения некоторых вероятностей, не обременяя пользователя лишними вопросами. Для этого необходимо конкретизировать *лингвистические переменные*. Пользователю экспертной системы нужно позволить добавлять к этим переменным определения, например «маленький» или «средний». Пользователь может задать маленькое повышение курса доллара и экспертная система должна точно знать, что под этим подразумевается.

КУ используется в области математики, называемой *нечеткой логикой*. КУ может иметь значение от -1 до 1 . Отрицательное значение КУ показывает степень уверенности в том, что правило не верно, а положительное значение – что верно. Правила, для которых $KU=-1$, рассматривать нет смысла.

Прежде всего сформулируем общие принципы вычисления КУ правила:

1. Выбрать максимальное значение КУ из КУ для условий правила, разделенных логическим оператором И.
2. Если в правиле есть оператор ИЛИ, выбрать максимальное значение из КУ для всех условий правила, разделенных оператором И для всех условий, связанных оператором ИЛИ.
3. Умножить выбранный КУ на КУ правила.
4. Если существует несколько правил с одинаковым логическим выводом, выбрать из всех полученных КУ максимальный.

Во многих случаях изначально заданы граничные значения коэффициента уверенности. Логический вывод считается верным только в том случае, если его КУ превышает заранее заданные граничные значения. Работа с базой знаний продолжается до тех пор, пока значение коэффициента уверенности логического вывода больше граничного значения. В процессе работы выполняются определенные вычисления.

Предположим, для частного логического вывода КУ равно 0,4. Это значение запоминается. Затем оно сравнивается с граничным значением КУ (допустим, что оно равно 0,8). Запомненное значение оказалось меньше граничного, и, значит, работа с базой знаний продолжается. Если при работе с базой знаний встретился тот же самый логический вывод, КУ для нового правила умножается на 1 минус значение запомненного ранее КУ и результат прибавляется к запомненному ранее КУ. Значение КУ, равное 1, свидетельствует об абсолютной уверенности в правильности вывода. Затем вновь запомненное значение КУ сравнивается с граничным и если оно больше, выполняется логический вывод, в противном случае, работа с базой знаний продолжается. Вышесказанное можно записать с помощью равенства:

$$\text{Запомненный КУ} = \text{Ранее запомненный КУ} + (1 - \text{Ранее запомненный КУ}) * \text{КУ нового правила}$$

Например:

Граничное значение КУ = 0,8

Правило: ЕСЛИ А, ТО В (КУ=0,6)

Запомненный КУ: 0,6

Новое правило: ЕСЛИ С, ТО В (КУ=0,7)

Запомненный КУ=0,6+ (1-0,6)*0,7=0,88 (граничные значения превышены, и выполняется вывод).

2. Порядок выполнения работы :

1. Для выбранного варианта написать правила, используя коэффициенты уверенности
2. Реализовать «дружественный интерфейс» пользователя;
3. Организовать вывод информации пользователю;
4. Описать программное обеспечение реализации задачи.

3. Варианты заданий

Описать правила и реализовать программный код, используя теорию нечеткой логики для следующих задач:

1. диагностика неисправностей электронной аппаратуры
2. диагностика неисправностей автомобиля
3. диагностика заболеваний (по выбору)
4. прогнозирование (по выбору)
 - a. спортивных мероприятий
 - b. телепередач
 - c. природных катаклизмови т.п.
5. классификация объектов (по выбору)
6. задачи информационно-советующего характера (по выбору)
 - a. помощник заведующего склада
 - b. помощник аптекаря
 - c. помощник оператора справочной службы
 - d. выбор должности
 - e. проведение отпускаи т.п.

4. Контрольные вопросы

1. Какие переменные называются лингвистическими?
2. В каких случаях используются КУ?
3. В каких пределах изменяется КУ?
4. В чём отличие создания правил при использовании КУ?

Практическая работа №5 Методика построения и архитектура экспертных систем

Краткие теоретические сведения

Под **экспертной системой (ЭС)** понимают набор программ, выполняющий функции эксперта при решении задач из некоторой предметной области. ЭС выдают советы, проводят анализ, дают консультации, ставят диагноз.

Главным достоинством ЭС, определяющим сравнительно высокий интерес к ним как к методам искусственного интеллекта, является возможность накопления знаний и сохранение их длительное время. В отличие от человека к любой информации ЭС подходят объективно, что улучшает качество проводимой экспертизы. При решении задач, требующих обработки большого объема знаний, возможность возникновения ошибки при переборе очень мала.

Структура традиционной **статической** ЭС включает следующие основные компоненты /1/ (рисунок 1):

- решатель (интерпретатор),
- рабочую память,
- базу знаний,
- компонент приобретения знаний,
- объяснительный компонент,
- диалоговый компонент.

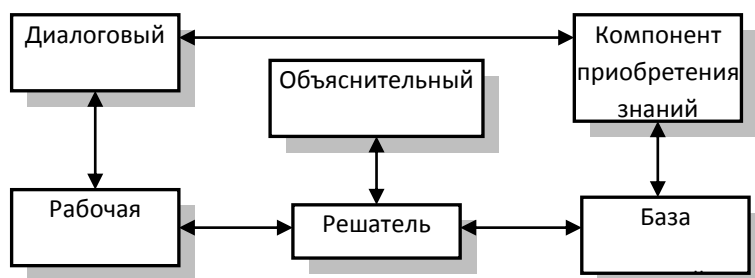


Рисунок 1 – Структура ЭС

В **базе знаний** содержатся факты, на основе которых производится выработка решения. **Решатель** – алгоритм, программа, набор правил, по которым осуществляется решение задачи. Процесс рассуждений реализуется на основе базы знаний и рабочей памяти. Решатель выполняет две функции: во-первых, просмотр существующих фактов из рабочей памяти и правил из базы знаний и добавление (по мере возможности) в рабочую память новых фактов и, во-вторых, определение порядка просмотра и применения правил. Одним из распространенных алгоритмов решателя является **байесовский алгоритм**, описанный в подразд. 3.2. На **диалоговый компонент** возложена задача ведения диалога о решаемой задаче на языке пользователя (эксперта). **Компонент приобретения знаний** как программный модуль может в ЭС отсутствовать. Его задача – приобретать в ходе диалога новые знания. Наличие **объяснительного компонента** дает ЭС способность при решении задачи следовать линии рассуждений, понятной пользователю (эксперту), и объяснять ход рассуждений.

Задание, порядок выполнения работы и содержание отчета

Работа выполняется в два этапа. При выполнении первого этапа лабораторной работы используется программа **Mini Expert System** («Малая экспертная система» вер. 1.0), описание работы с которой приведено ниже. Для второго этапа работы используется расширение **Fuzzy Logic Toolbox** пакета **MATLAB**.

Общее задание на лабораторную работу:

1 этап. В соответствии с вариантом составить список вопросов, необходимых для получения определенного решения и список вариантов ответов. Списки оформить в виде текстового файла с расширением .DAT (см. примеры из программы Mini Expert System).

Отладить экспертную систему, проверить ее работоспособность на примерах.

2 этап. Разработать экспертную систему оценки знаний экзаменуемого в соответствии с заданием. Для работы использовать расширение **Fuzzy Logic Toolbox** пакета **MATLAB** (описание приводится).

Содержание отчета по **первому** этапу работы:

- задание,
- списки вопросов и вариантов ответов (распечатка .DAT-файла),
- протоколы проверки работоспособности на примерах.

Содержание отчета по **второму этапу** работы:

- задание,
- продукционные правила разработанной экспертной системы,
- тип и интервалы определенности функций принадлежности входных и выходной переменной;
- протоколы проверки работоспособности на примерах.

Варианты заданий **на первый этап** работы:

Вариант 1. Идентификация типа транспортного средства (велосипед, мотоцикл, мотороллер, телега, карета, автобус, грузовик, легковые: пикап, седан, хэтчбек, кабриолет...).

Вариант 2. Проведение летнего отдыха (дома, в саду, в пешем походе, в местном санатории, на Черном море, на Средиземном море, в круизе на теплоходе, на горном курорте, в африканских странах и т.д.).

Вариант 3. Выбор принтера (или к.-л. другой техники по выбору) для покупки (матричного, струйного, лазерного).

6

Вариант 4. Где поужинать вечером? (дома, у друзей, в столовой, в кафе, в ресторане, в клубе).

Вариант 5. Выбор телевизора для дома (диагональ, тип, цена, марка и т.д.).

Вариант 6. Покупка квартиры в г. Уфе (цена, площадь, престижность района, экологическая ситуация в районе, транспорт, тип дома и т.д.).

Вариант 7. Идентификация заглавных букв греческого алфавита.

Вариант 8. Идентификация садовых растений (огурцы, томаты, лук, яблоня, вишня, смородина, крыжовник и т.д.).

Задание на **второй** этап работы.

Разработать экспертную систему оценки знаний по результатам ответа на экзаменационный билет. Билет содержит два теоретических вопроса и одно практическое задание. При неудовлетворительной оценке за практическое задание общая оценка за экзамен признается неудовлетворительной. Знания по каждому вопросу билета и на весь билет оцениваются по 4-бальной шкале – неудовлетворительно, удовлетворительно, хорошо, отлично.

Продукционная экспертная система в среде MATLAB

Перед началом разработки экспертной системы с использованием инструментальных средств **MATLAB** необходимо разработать продукционную систему (продукционные правила), которые будут использоваться при выводе. Правила имеют следующую структуру:

ЕСЛИ (1 воп. – хор) и (2 воп. – хор) и (3 воп. – хор), ТО (оцен. – хор.);

ЕСЛИ (1 воп.–хор) и (2 воп. –хор) и (3 воп.–неуд.), ТО (оцен. – неуд.);

ЕСЛИ (1 воп.–отл.) и (2 воп.–отл.) и (3 воп.–уд.), ТО (оцен. – хор.);

и т. д.

Вывод каждого правила определяется субъективными представлениями эксперта.

Разрабатываемая экспертная система оценки ответа на экзаменационный билет в среде расширения **Fuzzy Logic Toolbox** пакета **MATLAB** использует в своем выводе элементы теории нечетких множеств.

Редактор **Fuzzy Logic Toolbox** запускается командой **fuzzy** из командной строки пакета **MATLAB**. Новая система с выводами по Мамдани задается командой *File/New FIS/Mamdani*.

По заданию оценивается билет из трех вопросов, поэтому на входе системы должны быть заданы три входа. Входы добавляются последовательно (*Edit/Add Variable.../Input*) после конфигурирования каждого предыдущего входа. Конфигурирование заключается в определении

функции степени принадлежности, назначении интервалов определенности и названия.

В рассматриваемой задаче диапазон изменения входной переменной (поле *Range*, окно *Member Function Editor*) назначается от 0 до 5. Всего задаются 4 функции принадлежности. Каждой функции определяется название – n (неуд.), ud (удовл.), hor (хор.), ot (отл.). Тип функции принадлежности – трапецевидная (*trapmf*), с границами, например, как на рисунке 3. (можно задаться другим видом функции степени принадлежности).

Аналогично определяется выходная переменная. Вход в режим редактирования входной переменной производится двойным нажатием левой клавиши мыши. Добавление функции принадлежности – *Edit/Add MFs...*

Для наполнения базы знаний входят в режим ввода правил (*Edit/Rules*). Разработанные продукционные правила последовательно вводятся (*Add rule*) в систему вывода.

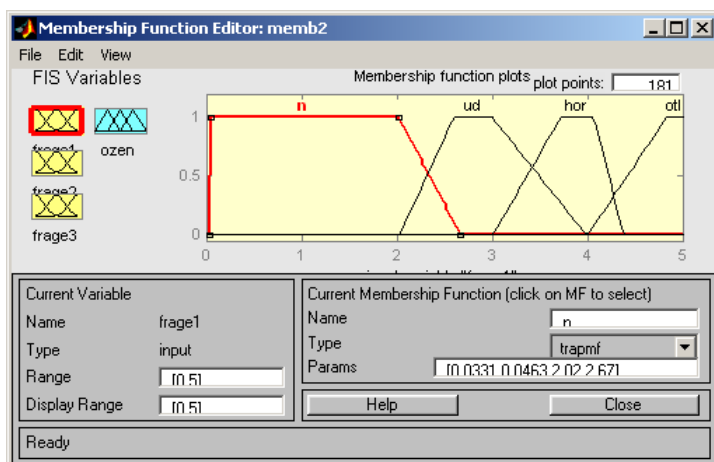


Рисунок 3 – Формы функций степени принадлежности

Для работы с созданной экспертной системой нужно из окна редактора правил (*Rule Editor*) перейти в окно вывода (*Rule Viewer*) *View/Rules*. Входными переменными является оценка за ответ каждого вопроса по 5-бальной шкале. Значения входных переменных вводятся в поле *Input*. Общая оценка ответа согласно вывода выводится в верхней части экране (синим цветом).

Контрольные вопросы

- 1 Что такое экспертная система?
 - 2 Как функционирует экспертная система? Какие функции выполняет каждый элемент системы?
 - 3 К какому типу относятся ядра продукций в разработанной Вами ЭС?
 - 4 Приведите пример одного продукционного правила, соответствующего разработанной базе знаний.
 - 5 Как осуществляется приобретение знаний в разработанной ЭС?
 - 6 В чем отличие однозначных и альтернативных продукционных правил?
 - 7 В чем достоинства и недостатки представления знаний в виде продукционных систем?
- .