

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Амурский государственный университет»

И.М. Акилова

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Методическое пособие

Благовещенск

2020

Системы искусственного интеллекта: методическое пособие для студентов направления подготовки 09.04.04 – Программная инженерия. /Составитель И. М. Акилова, – Благовещенск.: АмГУ, 2020 г. – 45 с.

Пособие содержит курс лекций, методические рекомендации по проведению и выполнению лабораторных работ. Составлено в соответствии с требованиями государственного образовательного стандарта.

## ВВЕДЕНИЕ

Основной особенностью интеллектуальных систем является то, что они основаны на знаниях, а вернее, на некотором их представлении. Знания в интеллектуальных системах понимаются как хранимая (с помощью ЭВМ) информация, формализованная в соответствии с некоторыми правилами, которую ЭВМ может использовать при логическом выводе по определенным алгоритмам. Наиболее фундаментальным и важным является описание смыслового содержания проблем самого широкого диапазона, т.е. должна использоваться такая форма описания знаний, которая гарантировала бы правильную обработку их содержимого по некоторым формальным правилам. Эта проблема называется проблемой представления знаний.

Типичными для представления знаний являются модели, основанные на использовании логики предикатов, продукций, фреймов и семантических сетей. Каждый способ представления знаний имеет свои достоинства и недостатки и тяготеет к определенной структуре знаний.

Практикум «Системы искусственного интеллекта» посвящен методам представления знаний с использованием продукций, фреймов, семантических сетей и логики предикатов.

Цель практикума – дать общее представление о моделях представления знаний и научить обрабатывать эти знания.

Каждая практическая работа содержит краткое описание прикладных систем и их особенностей. Рассмотрены также способы представления и использования нечетких знаний.

Практикум "Системы искусственного интеллекта" рекомендуется студентам специальности 09.04.04 – "Программная инженерия", изучающим курс "Системы искусственного интеллекта"

**ПРЕДСТАВЛЕНИЕ ЗНАНИЙ И ПОЛУЧЕНИЕ ВЫВОДОВ  
С ПОМОЩЬЮ ЛОГИКИ ПРЕДИКАТОВ**

Предикатом [2], или логической функцией, называется функция от любого числа аргументов, принимающая истинностные значения: истинно (1) и ложно (0). Аргументы принимают значения из произвольного, конечно-го или бесконечного множества  $D$ , называемого предметной областью. Предикат от  $n$  аргументов называют  $n$ -местным предикатом.

Предикат  $F(x)$ , определенный на множестве  $D$ , задает определенное свойство элементам множества  $D$  и интерпретируется как высказывание “ $x$  обладает свойством  $F$ ”, причем  $F$  принимает значение “истинно”, если это высказывание истинно, и значение “ложно”, если оно ложно. Предикат  $F(x_1, x_2, \dots, x_n)$  задает отношение между элементами  $x_1, x_2, \dots, x_n$  и интерпретируется как высказывание “ $x_1, x_2, \dots, x_n$  находятся между собой в отношении  $F$ ”.

Например:  $D$  – множество натуральных чисел.  $F(x)$  может означать, что  $x$  – четное или  $x$  – нечетное.  $G(x, y)$  –  $x > y$  или  $x$  делится на  $y$ .

Алфавит языка предикатов первого порядка состоит из:

- 1) разделители: запятая, открывающая и закрывающая скобки;
- 2) константы, обозначаемые строчными буквами или соединением таких букв, – например:  $a, st$ , друг;
- 3) переменные, обозначаемые прописными буквами, – например:  $X, ST$ , АДРЕС;
- 4) предикаты, обозначаемые прописными буквами, – например:  $P, Q$ , БОЛЬШЕ;
- 5) функции, устанавливающие зависимость и отображающие значения одной предметной области в значения другой (или той же),  $n$ -местные функции могут служить аргументами предиката. Функции будем обозначать строчными буквами:  $f, g, t$ ;
- 6) логические функции:

$\neg$  (отрицание или дополнение). Высказывание " $\neg A$ " читается "не А". Оно истинно (И), если высказывание А – ложно (Л);

$\wedge$  (конъюнкция). Высказывание " $A \wedge B$ " читается "А и В". Оно истинно в том случае, когда истинно как А, так и В;

$\vee$  (дизъюнкция). Высказывание " $A \vee B$ " читается "А или В". Оно истинно, если истинно хотя бы одно из высказываний;

$\rightarrow$  (импликация). Высказывание " $A \rightarrow B$ " читается "если А, то В". Оно ложно в том и только в том случае, если А истинно, а В ложно;

$\leftrightarrow$  (эквивалентность). Высказывание " $A \leftrightarrow B$ " читается "А тогда и только тогда, когда В". Оно истинно в тогда и только тогда, когда А и В имеют одно и тоже истинностное значение;

$\exists$  (квантор существования). Высказывание " $\exists A$ " читается "существует А";

$\forall$  (квантор общности). Высказывание " $\forall A$ " читается "для любого А".

*Пропозициональной формой* [2], или *формулой алгебры логики*, называют всякое высказывание, составленное из некоторых исходных высказываний посредством логических операций, т. е. если F и G – пропозициональные формы, то  $\neg F$ ,  $(F \vee G)$ ,  $(F \wedge G)$ ,  $(F \rightarrow G)$ ,  $(F \leftrightarrow G)$  – пропозициональные формы.

*Пропозициональной функцией* [5] называется выражение, содержащее переменную и превращающееся в истинное или ложное высказывание при подстановке вместо переменной имени предмета из определенной предметной области.

Пропозициональные функции делятся на одноместные, содержащие одну переменную, называемые свойствами (например, «х – композитор», «х-7=3»), и содержащие две и более переменных, называемые отношениями (например, «х-с=16», «объем куба х равен объему куба у).

Определение формулы – основного объекта логики предикатов – включает понятие "терм".

*Терм* – выражение, включающее константы, переменные или  $n$ -местные функции  $f(t_1, t_2, \dots, t_n)$ , где  $t_1, t_2, \dots, t_n$  – термы.

Например:  $f(X, Y)$ ,  $f(b, \text{вес}(Z))$  – термы,  $P(X, \text{голубой})$ ,  $\text{вес}(P(b))$  – не термы, т.к.  $P$  – предикат.

*Атом*, или *элементарная (атомная) формула* – это выражение, включающее константы, переменные, функции и предикаты. Таким образом, если  $P$  –  $n$ -местный предикат, а  $t_1, t_2, \dots, t_n$  – термы, то  $P(t_1, t_2, \dots, t_n)$  – атом.

Например:  $P(X, \text{голубой})$ ,  $\text{ВХОД}(\text{стол}, X, \text{под}(\text{окно}))$  – являются атомами,  $f(X, Y)$  – не атом.

Формула или правильно построенная формула (ППФ) определяется следующим образом: всякий атом есть ППФ.

Если  $F$  и  $G$  – ППФ, а  $X$  – переменная, тогда  $\neg H$ ,  $(G \vee H)$ ,  $(G \wedge H)$ ,  $(\exists X)G$ ,  $(\forall X)H$  – ППФ.

Выражение “первого порядка” во фразе “исчисление предикатов первого порядка” связано с определением ППФ, в которых запрещается квантифицировать символы предикатов и функций.

Например:

$(\forall P)P(a)$   
 $(\forall f)(\forall X)P(f(X), b)$  – не являются ППФ логики предикатов первого порядка.

На практике ППФ используется для представления знаний. Не всегда легко представить знания, выраженные на естественном языке, с помощью ППФ. Например, выражение “если два объекта равны, то они имеют одинаковые свойства” можно представить так;

$$(\forall P)(\forall X)(\forall Y)(P \text{ РАВНО}(X, Y) \rightarrow (P(X) \leftrightarrow P(Y))).$$

Но это выражение не является формулой первого порядка, так как квантифицируется предикат.

Покажем, каким образом можно выявлять логическую структуру мысли. Приведем несколько сложных суждений, структуру которых надо выразить в виде формул, используя введенные логические термины.

1. Если у меня будет свободное время (а) и я сдам экзамены по математике (b) и физике (с), то поеду отдыхать в Крым (d) или на Кавказ (е).

Формула:  $(a \wedge b \wedge c) \rightarrow (d \vee e)$ .

2. «Если человек с детства и юности своей не давал нервам властвовать над собой, то они не привыкнут раздражаться и будут ему послушны» (К.Д.Ушинский).

Формула:  $(\neg a \wedge \neg b) \rightarrow (\neg c \wedge d)$ .

Здесь буква  $a$  обозначает суждение: «человек с детства давал нервам властвовать над собой». А так как у нас имеется отрицание («не давал»), то запишем  $\neg a$ .

3. Если добродетель неправильно приложат, то она может стать пороком.

Формула:  $a \rightarrow b$ .

4. Если у меня будет свободное время, то я почитаю книгу или посмотрю телевизор.

Формула:  $a \rightarrow (b \vee c)$ .

5. Некоторые элементарные частицы имеют положительный заряд.

Формула:  $(\exists X)(S(X) \wedge P(X))$ .

*Правилом вывода* называют процедуру, которая из одной или нескольких ППФ производит другие ППФ. Задавая фиксированное множество правил вывода, можно рассматривать следующее семейство проблем: исходя из выбранного множества ППФ применением некоторого числа раз правил вывода, можно получить заранее заданную ППФ.

Исходные ППФ называют *аксиомами*, а ППФ, полученные из правил вывода, называют *теоремами*. Цель применения правил вывода, ведущих от аксиомы к теореме, называют *доказательством теоремы*.

*Интерпретация формул.* Формула имеет определенный смысл, т. е. обозначает некоторое высказывание, если существует какая-либо интерпретация. Интерпретировать формулу – это значит связать с ней определенное

непустое множество  $D$ , т. е. конкретизировать предметную область, называемую также *областью интерпретации*, и указать:

для каждой константы в формуле – конкретный элемент из  $D$ ;

для каждой  $n$ -местной функциональной буквы в формуле – конкретную  $n$ -местную функцию на  $D$ ;

для каждой  $n$ -местной предикатной буквы в формуле – конкретное отношение между  $n$  элементами из  $D$ .

*Свойства правильно построенных формул.* При заданной интерпретации значения истинности правильно построенную формулу (ППФ) можно вычислить по правилам, объединенным в таблице истинности.

Если  $F$  и  $G$  – любые две ППФ, то значения истинности составного выражения, построенного из этих ППФ, даются таблицей истинности (табл. 1).

Таблица 1. Таблица истинности

F	G	$\neg F$	$F \vee G$	$F \wedge G$	$F \rightarrow G$	$F \leftrightarrow G$
И	И	Л	И	И	И	И
Л	И	И	И	Л	И	Л
И	Л	Л	И	Л	Л	Л
Л	Л	И	Л	Л	И	И

Если истинности двух ППФ независимо от интерпретации совпадают, то говорят, что эти ППФ являются эквивалентными. Пользуясь таблицей истинности, легко установить следующие эквивалентности (табл. 2.).

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что называется предикатом?
2. Из чего состоит алфавит языка предикатов первого порядка?
3. Что называют формулой алгебры логики?
4. Что такое элементарная формула?
5. Какую процедуру называют правилом вывода?
6. Что значит интерпретировать правильно построенную формулу?

Таблица 2. Таблица эквивалентности

Эквивалентные формулы		
$\neg(\neg F)$	$F$	
$F \rightarrow G$	$\neg F \vee G$	
$F \leftrightarrow G$	$(F \rightarrow G) \wedge (G \rightarrow F)$	
$\neg(F \wedge G)$	$\neg F \vee \neg G$	Законы де Моргана
$\neg(F \vee G)$	$\neg F \wedge \neg G$	
$F \wedge (G \vee H)$	$(F \wedge G) \vee (F \wedge H)$	Законы дистрибутивности
$F \vee (G \wedge H)$	$(F \vee G) \wedge (F \vee H)$	
$F \vee G$	$G \vee F$	Законы коммутативности
$F \wedge G$	$G \wedge F$	
$(F \wedge G) \wedge H$	$F \wedge (G \wedge H)$	Законы ассоциативности
$(F \vee G) \vee H$	$F \vee (G \vee H)$	
$F \rightarrow G$	$\neg G \rightarrow \neg F$	
$F \wedge \neg F$	$0$	
$F \vee \neg F$	$1$	
$F \wedge 0$	$0$	
$F \wedge 1$	$F$	
$F \vee 0$	$F$	
$F \vee 1$	$1$	
$\forall X F(X)$	$\forall Y F(Y)$	
$\exists X F(X)$	$\exists Y F(Y)$	
$\neg(\exists X F(X))$	$\forall X(\neg F(X))$	
$\neg(\forall X F(X))$	$\exists X(\neg F(X))$	
$\forall X(F(X) \wedge G(X))$	$\forall X F(X) \wedge \forall X G(X)$	
$\exists X(F(X) \vee G(X))$	$\exists X F(X) \vee \exists X G(X)$	

### ЗАДАНИЕ

Выразить в символической форме следующие сложные суждения:

1. Если встать рано на рассвете и пойти в сад или парк, то можно услышать чудесные песни птиц.
2. Если эта фигура квадрат, то диагонали в ней равны, взаимно перпендикулярны и в точке пересечения делятся пополам.
3. Некоторые спортсмены не являются мастерами спорта.
4. «Некоторые лекарства опаснее самих болезней» (Сенека).
5. Некоторые люди не изучают логику.
6. Если мне дадут отпуск летом, то я поеду отдыхать к морю или по туристической путевке в Канаду.
7. Добро не умрет, а зло пропадет.
8. «Видеть несправедливость и молчать – это значит самому участвовать в ней» (Ж.-Ж. Руссо).
9. Если вы любите детей, полны жажды познания, имеете доброе сердце, мечтаете посвятить себя интересному творческому труду, то смело выбирайте профессию учителя.
10. «Если больному после разговора с врачом не становится легче, то это не врач» (В. М. Бехтерев).
11. «Если верный конь, поранив ногу,  
Вдруг споткнулся, а потом опять,  
не вини коня, вини дорогу  
и коня не торопись менять» (Р. Гамзатов).

## *ПРАКТИЧЕСКАЯ РАБОТА № 2*

### ***ПРЕОБРАЗОВАНИЕ ПРАВИЛЬНО ПОСТРОЕННЫХ ФОРМУЛ В ПРЕДЛОЖЕНИЕ***

Аппарат логики предикатов используется для представления задачи в виде теоремы: формула  $H$  логически следует из формулы  $G$ , т.е.  $G \rightarrow H$ . Доказательство этой теоремы состоит в том, чтобы показать, что каждая интерпретация, удовлетворяющая  $G$ , удовлетворяет и  $H$ , или, что  $\neg G \vee H$  ис-

тинно для любой интерпретации; так как  $\neg(\neg G \vee H) = G \wedge \neg H$ , то на практике обычно доказывают невыполнимость множества предложений  $G \wedge \neg H$ .

Чтобы упростить доказательства теоремы, все формулы представляются в виде дизъюнкций литералов. *Литералом* (или *литерой*) называют атом или его отрицание. Формулу, представляющую собой дизъюнкцию литералов, называют *предложением* (или *дизъюнктом*).

Например:  $R(Z, a, g(X)) \vee (\neg T(U)) \vee (\neg U(b, k(c)))$ .

Иными словами, в каждой формуле необходимо исключить все логические операции (включая кванторы), кроме дизъюнкции, и уменьшить область действия знака отрицания до одной предикатной буквы.

*Преобразование ППФ в предложение.* Перед тем как объяснить процесс резолюции, покажем, что любую ППФ исчисления предикатов первого порядка можно преобразовать во множество предложений, используя законы эквивалентности. Процесс такого преобразования состоит из последовательных этапов, которые покажем на следующем примере ППФ:

$(\forall X)((P(X) \wedge Q(X, a)) \rightarrow R(X, b)) \wedge ((\forall Y)((\forall Z)(R(Y, Z) \rightarrow T(X, Y)))) \vee ((\forall X)S(X))$ .

*Исключение символов эквивалентности и импликации.* Для этого воспользуемся следующими законами эквивалентности:  $(F \rightarrow G)$  заменим на  $(\neg F \vee G)$ . В нашем примере такая подстановка даст:

$(\forall X)((\neg(P(X) \wedge Q(X, a)) \vee R(X, b)) \wedge ((\forall Y)(\neg(\forall Z)(R(Y, Z) \vee T(X, Y)))) \vee ((\forall X)S(X))$ .

*Уменьшение области действия знаков отрицания.* Нужно выполнить такое преобразование, чтобы каждый знак отрицания применялся не более чем к одной атомной формуле. Для этого используем следующие законы эквивалентности:

$(\neg(\neg F))$  и  $F$

$(\neg(F \wedge G))$  и  $\neg F \vee \neg G$

$(\neg(F \vee G))$  и  $\neg F \wedge \neg G$

$(\neg(\exists X)F(X))$  и  $(\forall X)(\neg F(X))$

$$(\neg(\forall X)F(X)) \text{ и } (\exists X)(\neg F(X))$$

Полученная нами ППФ преобразуется к виду:

$$(\forall X)((\neg P(X) \vee \neg Q(X,a) \vee (R(X,b))) \wedge ((\forall Y)((\exists Z)(\neg R(Y,Z)) \vee T(X,Y)))) \vee ((\forall X)S(X))$$

*Разделение переменных.* В пределах области действия квантора, переменная, связываемая с этим квантором, представляет собой немую переменную. Такую переменную в области действия квантора можно заменить любой другой (не встречающейся) переменной, при этом значение истинности ППФ не изменится. Стандартизация переменных в пределах ППФ означает переименование немых переменных с той целью, чтобы каждый квантор имел свою, свойственную только ему, немую переменную. Так, вместо записи

$$(\forall X)(P(X) \vee (\exists X)Q(X))$$

можно написать

$$(\forall X)(P(X) \vee (\exists Y)Q(Y)).$$

Полученная на предыдущем этапе ППФ преобразуется к виду:

$$(\forall X)((\neg P(X) \vee \neg Q(X,a) \vee (R(X,b))) \wedge ((\forall Y)((\exists Z)(\neg R(Y,Z)) \vee T(X,Y)))) \vee ((\forall U)S(U))$$

*Исключение кванторов существования.* Рассмотрим ППФ

$$(\forall X)(\exists Y)P(X,Y),$$

которую можно прочесть как: "Для всех  $X$  существует некоторое  $Y$  (возможно, зависящее от  $X$ ) такое, что  $P(X,Y)$ ". Поскольку квантор существования находится в области некоторого квантора общности, то можно допустить, что "существующий"  $Y$  зависит от  $X$ . Пусть эта зависимость определяется функцией  $Y=g(X)$ , отображающей каждое  $X$  в  $Y$ . Такая функция называется *сколемовской*, или *функцией Сколема*. Заменяв  $Y$  на  $g(X)$ , мы исключим квантор существования, и наша ППФ примет вид:

$$(\forall X)(P(X,q(X))).$$

Пусть ППФ  $(\exists Y)P(Y)$  является подформулой другой ППФ, в которой  $X_1, \dots, X_n$  квантифицированы универсально. Тогда удаляют  $Y$  и заменяют встречающуюся  $Y$  сколемовской функцией  $f(X_1, \dots, X_n)$ . Заметим, что эта функция будет содержать столько аргументов, сколько имеется универсаль-

ных квантификаторов слева от формулы  $(\exists Y)P(Y)$ . Эта функция, как мы уже говорили, просто выражает существование соответствия априори между совокупностью значений  $(\exists Y)P(Y)$  и значениями  $Y$ . Поскольку эта функция заранее неизвестна, то в каждой замене  $Y$  мы должны записывать новую функциональную букву, которая отличается от уже имеющихся. Когда слева от символа "Э" отсутствуют символы "В", то вводимая функция Сколема не будет содержать аргументов: следовательно, это будет новая константа, называемая константой Сколема. Так, ППФ  $(\exists Y)P(Y)$  заменится на  $P(a)$ , где про константу "a" известно, что она существует.

После выполнения замены  $Z$  функцией Сколема  $g(X, Y)$  наша ППФ будет иметь вид

$$(\forall X)((\neg P(X) \vee \neg Q(X, a) \vee (R(X, b) \wedge ((\forall Y)(\neg R(Y, g(X, Y)) \vee T(X, Y)))))) \vee ((\forall U)S(U)).$$

*Перемещение всех квантификаторов общности в начало ППФ* (без изменения их относительного порядка). Каждый квантор общности имеет свою переменную, и поэтому такие кванторы можно переместить в начало формулы, считая, что область действия каждого из них включает всю последующую часть ППФ. Говорят, что результирующая ППФ находится в префиксной форме. Она состоит из цепочки кванторов, называемой *префиксом*, и следующей за ней бескванторной формулы, называемой *матрицей*.

Префиксная форма для нашей ППФ имеет вид:

$$(\forall X)(\forall Y)(\forall U)((\neg P(X) \vee \neg Q(X, a) \vee R(X, b) \wedge (\neg R(Y, Z) \vee T(X, Y))) \vee S(U)).$$

*Приведение матрицы к конъюнктивной нормальной форме.* Матрица находится в *конъюнктивной нормальной форме*, если она записана как конъюнкция конечного множества дизъюнкций литералов. Например:

$$\begin{aligned} &(P(X) \vee Q(X, Y)) \wedge (P(Y) \vee \neg R(Y)) \\ &Q(X) \wedge R(X, Y) \\ &\neg R(Y) \end{aligned}$$

Любую матрицу можно привести к конъюнктивной нормальной форме, используя законы ассоциативности и дистрибутивности операций  $\vee$  и  $\wedge$ .

После приведения матрицы нашего примера ППФ в конъюнктивную нормальную форму эта ППФ примет следующий вид:

$$(\forall X)(\forall Y)(\forall U)(\neg P(X) \vee \neg Q(X,a) \vee (R(X,b) \vee S(U))) \wedge (\neg R(Y,g(X,Y)) \vee T(X,Y) \vee S(U)).$$

*Исключение кванторов общности.* Поскольку все переменные в ППФ должны быть связаны, то можно предположить, что для каждой из них имеется квантор общности. При этом порядок расположения этих кванторов роли не играет, так что можно их исключить из ППФ. После исключения кванторов из ППФ у нас остается лишь матрица в конъюнктивной нормальной форме.

*Исключение символов конъюнкции.* Теперь можно исключить символы  $\wedge$ , путем замены выражения типа  $(P \wedge Q)$  на множество ППФ  $\{P, Q\}$ . В результате выполнения таких замен получим конечное множество ППФ, каждая из которых – дизъюнкция литералов. А любая ППФ, содержащая дизъюнкцию литералов, как мы условились, является предложением.

Наш пример ППФ преобразуется в следующее множество предложений:

$$\{\neg P(X) \vee \neg Q(X,a) \vee (R(X,b) \vee S(U))\} \\ \{\neg R(Y,g(X,Y)) \vee T(X,Y) \vee S(U)\}.$$

*Переименование (разделение) переменных.* Символы переменных должны быть изменены так, чтобы каждый появился не более чем в одном предложении. Напомним, что  $(\forall X(P(X) \wedge Q(X)))$  эквивалентно  $(\forall X)P(X) \wedge (\forall Y)Q(Y)$ . Теперь наши предложения приобретают вид:

$$\neg P(X) \vee \neg Q(X,a) \vee (R(X,b) \vee S(U)) \\ \neg R(Y,g(Z,Y)) \vee T(A,B) \vee S(C)$$

Вместо переменных в литералах могут находиться термы, не содержащие переменных. Тогда говорят, что имеет место основной частный случай. Например,  $P(a, f(g(b)))$  является основным частным случаем  $P(X, Y)$ .

По структуре высказывания делятся на простые (они имеют логическую форму «S есть P» либо «S не есть P») и сложные (грамматически выражаются сложными предложениями). Простые высказывания бывают

атрибутивные (в них выражается принадлежность или непринадлежность какого-то свойства объекту или классу объектов);  
об отношениях между несколькими объектами;  
о существовании или несуществовании какого-либо объекта или явления).

В атрибутивные высказывания часто включаются кванторные связки.

По качеству простые высказывания делятся на утвердительные и отрицательные. С количественной точки зрения высказывания делятся на единичные, частные и общие.

Перечислим основные типы высказываний и их обозначения [5]:

**A** – общеутвердительные («*Всякий S есть P*»),

**E** – общеотрицательные («*Всякий S не есть P*»),

**I** – частноутвердительные («*Некоторый S есть P*»),

**O** – частноотрицательные («*Некоторый S не есть P*»).

В частных суждениях слово «некоторые» не исключает варианта «все».

#### Пример.

Преобразовать высказывание "Все дети имеют матерей" в предложение, т.е. в дизъюнкцию литералов. Это высказывание можно представить следующей ППФ:

$$\forall X \exists Y (\text{ребенок}(X) \rightarrow \text{мать}(Y) \wedge \text{относится}(X, Y)).$$

Для удаления квантора существования заменим переменную Y функцией Сколема  $g(A)$ , т.е.  $Y=g(X)$ , откуда получим:

$$\forall X (\text{ребенок}(X) \rightarrow \text{мать}(g(X)) \wedge \text{относится}(X, g(X))).$$

Теперь можно исключить квантор общности, который распространяет свое действие на все выражения. Кроме того, заменим символ "импликация" дизъюнкцией, используя закон  $P \rightarrow Q \equiv \neg P \vee Q$ , откуда получаем:

$$\neg(\text{ребенок}(X) \vee (\text{мать}(g(X)) \wedge \text{относится}(X, g(X)))).$$

Используя закон

$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R),$$

можем написать

$$\neg(\text{ребенок}(X) \vee \text{мать}(g(X))),$$

$$\neg(\text{ребенок} \vee \text{относится}(X, g(X))).$$

Два полученные предложения можно интерпретировать так: "Если X – ребенок, то он относится к матери, выражаемой функцией  $g(X)$ ". Из них следует:

$\text{ребенок}(X) \rightarrow \text{мать}(g(X))$  и

$\text{ребенок}(X) \vee \text{относится}(X, g(X))$ .

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение литерала.
2. Что называют дизъюнктом?
3. Какая функция называется функцией Сколема?
4. Как любую ППФ можно преобразовать во множество предложений?
5. Когда ППФ находится в префиксной форме?
6. Какая ППФ называется матрицей?
7. Когда матрица находится в конъюнктивной нормальной форме?

## ЗАДАНИЕ

Преобразовать высказывания в дизъюнкцию литералов.

1. Некоторые растения являются грибами.
2. Все предложения со второстепенными членами являются распространенными.
3. Все одноклеточные не являются червяками.
4. Некоторые космонавты – летчики.
5. Некоторые студенты – не лодыри.
6. Все дельфины – теплокровные.

7. Ни один кит не является рыбой.
8. Все свидетели дают истинные показания.
9. Ни одна балалайка не является клавишным инструментом.
10. Некоторые люди не любят природу.

### *ПРАКТИЧЕСКАЯ РАБОТА № 3*

### **СУЩНОСТЬ ПРИНЦИПА РЕЗОЛЮЦИЙ.**

### **ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ПРИНЦИПА РЕЗОЛЮЦИЙ**

*Резольвенты* [2]. Говорят, что литерал является *конкретным*, если он не содержит никакой переменной. Например,  $\neg P(a)$  или  $Q(a, f(b))$  являются конкретными литералами, в то время как  $\neg P(X)$  или  $Q(X, f(y))$  не являются таковыми. Конкретным предложением является дизъюнкция конкретных литералов.

Пусть имеются два конкретных предложения

$$P_1 \vee P_2 \vee \dots \vee P_n \text{ и } \neg P_1 \vee Q_2 \vee \dots \vee Q_n$$

Здесь литерал  $\neg P_1$  является отрицанием литерала  $P_1$ . Из этих двух предложений можно вывести новое предложение, называемое *резольвентой*, или *резольвцией*. Резольвентой является дизъюнкция этих предложений с последующим исключением пары  $P_1$  и  $\neg P_1$ .

Из предложений  $P_1$  и  $\neg P_1$  получается *пустое предложение*, которое является признаком противоречия и обозначается символом  $\square$ .

Два конкретных предложения могут не иметь резольвенты или могут иметь множество резольвент. Например, из  $P \vee Q \vee R$  и  $\neg P \vee \neg Q \vee S$  получаются резольвенты  $Q \vee \neg Q \vee R \vee S$  или  $P \vee \neg P \vee R \vee S$  (которые в действительности являются эквивалентными).

*Сущность принципа резолюций.* На практике невыполнимость множества предложений устанавливается посредством принципа резолюций. Речь

идет о процедуре логического вывода новых предложений из множества исходных.

Принцип резолюций состоит:

1) в получении новых предложений на основании множества исходных и вновь получаемых предложений;

2) в отыскании частных случаев формул  $F(t_1, \dots, t_n)$  из  $F(V_1, \dots, V_n)$  при подстановке вместо  $V_i$  произвольных термов  $t_i$ , т.е.  $F(V_1, \dots, V_n) \rightarrow F(t_1, \dots, t_n)$ .

Резольвенту двух предложений можно получить следующим образом:

1) переименовать переменные двух предложений так, чтобы последние стали одинаковыми;

2) найти подстановку, преобразующую литерал одного предложения в дополнительный по отношению к некоторому литералу другого предложения и произвести эту замену в обоих предложениях;

3) вычеркнуть дополнительные друг к другу литералы;

4) удалить одинаковые литералы в предложении кроме одного;

5) дизъюнкция литералов, оставшихся в обоих предложениях, является резольвентой.

Если некоторая последовательность резолюций, применяемых к исходному множеству предложений  $E$  и множеству резольвент, полученных в процессе резолюции, приводит к пустому предложению, то множество  $E$  является *невыполнимым*.

Для доказательства невыполнимости множества предложений пользуются *опровержением*. Доказательство выполнимости множества  $G \rightarrow H$ , что эквивалентно  $\neg G \vee H$ , – это опровержение его невыполнимости, или, что то же самое, доказательство невыполнимости  $\neg(\neg G \vee H)$ , что эквивалентно  $G \wedge \neg H$ .

Общий алгоритм опровержения с помощью резолюций может иметь следующий вид:

**резолюция (G,H)**

- 1) С: – множество предложений, полученных путем преобразования формул множества G;
- 2) добавить к множеству С предложения, полученные из  $\neg H$ ;
- 3) пока пустое предложение не появится в С, **выполнить:**  
**начало:** выбрать два различных предложения в С;  
**если** они имеют резольвенты, то найти одну резольвенту и добавить ее к множеству С.

**конец.**

*Примеры использования метода резолюций.*

Пример 1. Предположим, что покупательная способность людей падает, если цены растут на товары и услуги. Предположим также, что люди несчастны, когда их покупательная способность падает. Предположим, что цены растут. Из этого можно заключить, что люди несчастны.

Применим следующие обозначения:

- 1) P – цены растут;
- 2) S – покупательная способность уменьшается;
- 3) U – люди несчастны.

В этом примере четыре утверждения:

- 1) если цены растут, то покупательная способность падает;
- 2) если покупательная способность падает, то люди несчастны;
- 3) цены растут;
- 4) люди несчастны.

Эти утверждения можно записать в виде следующих формул:

- 1)  $P \rightarrow S$ ,
- 2)  $S \rightarrow U$ ,
- 3) P,
- 4) U.

Преобразуем эти формулы в предложения:

- 1)  $\neg P \vee S$ ,
- 2)  $\neg S \vee U$ ,

3) P,

4) U.

Докажем путем опровержения, что U – логическое следствие из 1), 2), 3). Отрицаем 4) и получаем следующее доказательство:

1)  $\neg P \vee S$ ,

2)  $\neg S \vee U$ ,

3) P,

4)  $\neg U$  отрицание заключения,

5) S резольвента 3) и 1),

6) U резольвента 5) и 2),

7)  $\square$  резольвента 6) и 4).

Пример 2. Все доктора имеют пациентов и некоторые пациенты любят своих докторов. Ни один пациент не любит знахаря. Следовательно, никакой доктор не является знахарем.

Обозначим:

1) P(X): X – пациент;

2) D(X): X – доктор;

3) Q(X): X – знахарь;

4) L(X,Y): X любит Y.

Тогда факты и заключения можно записать следующим образом:

F1:  $(\exists X)(P(X) \wedge (\forall Y)(D(Y) \rightarrow L(X,Y)))$ ,

F2:  $(\forall X)(P(X) \rightarrow (\forall Y)(Q(Y) \rightarrow \neg L(X,Y)))$ ,

G:  $(\forall X)(D(X) \rightarrow \neg Q(X))$ .

Преобразуем эти формулы в предложения:

1) P(a) из F1,

2)  $\neg D(Y) \vee L(a,Y)$  из F1,

3)  $\neg P(X) \vee \neg Q(Y) \vee \neg L(X,Y)$  из F2,

4) D(b) из G,

5) Q(b) из G.

Используя метод резолюций, получим следующее доказательство:

- 6)  $L(a,b)$                       резольвента 4) и 2),
- 7)  $\neg Q(Y) \vee \neg L(a,Y)$       резольвента 3) и 1),
- 8)  $\neg L(a,b)$                       резольвента 5) и 7),
- 9)  $\square$                               резольвента 6) и 8).

Поясним естественность этого доказательства.

Из F1 можно предположить, что существует пациент  $a$ , который любит каждого доктора 1) и 2).

Предположим, что заключение неверно, т.е.  $b$  – одновременно и доктор, и знахарь.

Так как пациент любит каждого доктора, то  $a$  любит  $b$  6).

Так как  $a$  – пациент, то  $a$  не любит никакого знахаря 7).

Однако  $b$  – знахарь. Следовательно,  $a$  не любит  $b$  8).

Это невозможно из-за 6). Таким образом, мы закончили доказательство.

Пример 3. Допустим, что если Верховный Совет отказывается принимать новые законы, то забастовка не будет закончена, если только она не длится более года и зарплата не повышается. Закончится ли забастовка, если Верховный Совет отказывается действовать и забастовка только началась?

Применим следующие обозначения:

- 1)  $P$  – Верховный Совет отказывается действовать;
- 2)  $Q$  – забастовка заканчивается;
- 3)  $R$  – зарплата повышается;
- 4)  $S$  – забастовка длится более года.

Тогда приведенные выше утверждения можно представить следующими формулами:

- 1) F1:  $(P \rightarrow (\neg Q \vee (R \wedge S)))$  – если Верховный Совет отказывается принимать новые законы, то забастовка не будет закончена, если она не длится более года и зарплата не повышается;
- 2) F2:  $P$  – Верховный Совет отказывается действовать;

3) F3:  $\neg S$  – забастовка только началась.

Требуется доказать, что  $\neg Q$  – логическое следствие  $F1 \wedge F2 \wedge F3$ . Отрицаем Q и преобразуем F1, F2 и F3 в предложения:

- 1)  $\neg P \vee \neg Q \vee R$  из F1,
- 2)  $\neg P \vee \neg Q \vee S$  из F1,
- 3) P из F2,
- 4)  $\neg S$  из F3,
- 5) Q отрицание заключения.

Используя резолюции, получим следующее доказательство:

- 6)  $\neg Q \vee S$  резольвента из 3) и 2),
- 7) S резольвента из 6) и 5),
- 8)  $\square$  резольвента из 7) и 4).

Пример 4. Таможенники обыскивают каждого, кто въезжает в страну, кроме высокопоставленных лиц. Некоторые люди, способствующие провозу наркотиков, въезжали в страну и были обысканы людьми, также способствующими провозу наркотиков. Никто из высокопоставленных лиц не способствовал провозу наркотиков. Доказать, что некоторые из таможенников способствовали провозу наркотиков.

Примем следующие обозначения:

- 1) E(X): X въезжал в страну;
- 2) V(X): X – высокопоставленное лицо;
- 3) S(X,Y): Y обыскивает X;
- 4) C(X): X – таможенник;
- 5) P(X): X способствует провозу наркотиков.

Заключение представляется формулой:

$$(\exists X)(P(X) \wedge C(X)).$$

Отрицаем заключение и преобразуем формулы в предложения:

- 1)  $\neg E(X) \vee V(X) \vee S(X, f(X)),$
- 2)  $\neg E(X) \vee V(X) \vee C(f(X)),$

- 3)  $P(a)$ ,
- 4)  $E(a)$ ,
- 5)  $\neg S(a, Y) \vee P(Y)$ ,
- 6)  $\neg P(X) \vee \neg V(X)$ ,
- 7)  $\neg P(X) \vee \neg C(X)$ .

Доказательство методом резолюций выглядит следующим образом:

- 8)  $\neg V(a)$  из 3) и 6),
- 9)  $V(a) \vee C(f(a))$  из 2) и 4),
- 10)  $C(f(a))$  из 8) и 9),
- 11)  $V(a) \vee S(a, f(a))$  из 1) и 4),
- 12)  $S(a, f(a))$  из 8) и 11),
- 13)  $P(f(a))$  из 12) и 5),
- 14)  $\neg C(f(a))$  из 13) и 7)
- 15)  $\square$  из 10) и 14).

Заключение доказано.

Пример 5. Каждый, кто хранит деньги в кассе, получает проценты.

Докажем, что если не существует процентов, то никто не хранит денег.

Примем следующие обозначения:

- 1)  $S(X, Y)$ : X хранит деньги Y;
- 2)  $M(X)$ : X есть деньги;
- 3)  $I(X)$ : X есть проценты;
- 4)  $E(X, Y)$ : X получает Y проценты.

Посылка записывается в виде:

$$(\forall X)(\exists Y)(S(X, Y) \wedge M(Y)) \rightarrow (\exists Y)(I(Y) \wedge E(X, Y)),$$

а заключение в виде:

$$\neg(\exists X)I(X) \rightarrow (\forall X)(\forall Y)(S(X, Y) \rightarrow \neg M(Y)).$$

Преобразуем посылку в предложения:

- 1)  $\neg S(X, Y) \vee \neg M(Y) \vee I(f(X))$ ,
- 2)  $\neg S(X, f(X)) \vee \neg M(Y) \vee E(X, f(X))$ .

Преобразованное в предложение отрицание заключения имеет

вид:

3)  $\neg I(z)$ ,

4)  $S(a,b)$ ,

5)  $M(b)$ .

Доказательство заключения:

6)  $\neg S(X,Y) \vee \neg M(Y)$  из 3) и 1),

7)  $\neg M(b)$  из 6) и 4),

8)  $\square$  из 7) и 5).

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем смысл процедуры резолюции?
2. Что такое резольвента?
3. Какое предложение называется пустым?

### ЗАДАНИЕ

Используя метод резолюций, доказать истинность заключения.

1. Лошадь есть животное, поэтому голова лошади есть голова животного.
2. Полиция обыскивает всех въехавших в страну, за исключением дипломатов. Шпион въехал в страну, однако распознать личность шпиона может только шпион. Шпион не является дипломатом. Среди полицейских имеется шпион.
3. Благородный труд заслуживает уважения, так как благородный труд способствует прогрессу общества. Труд учителя есть благородный труд, так как труд учителя заключается в обучении и воспитании подрастающего поколения. Труд учителя заслуживает уважения.
4. Все тюльпаны – цветы. Все цветы – растения. Все растения используют для питания углекислый газ атмосферы и выделяют в нее кислород. Все растения, использующие для питания углекислый газ атмосферы и выделя-

ющие в нее кислород, содержат хлорофилл. Все тюльпаны содержат хлорофилл.

5. Все, что требует мужества и героизма, есть подвиг. Первый полет человека в космос требовал мужества и героизма. Первый полет человека в космос есть подвиг.

6. Если животное млекопитающее, то оно относится к типу хордовых. Это животное не является млекопитающим. Это животное не относится к типу хордовых.

7. Всякое преступление карается законом, поскольку оно общественно опасно. Грабеж есть преступление, так как грабеж – это открытое хищение личного имущества граждан. Грабеж карается законом.

8. Если должностное лицо получает взятку, то оно совершает преступление. Должностное лицо не получает взятку. Данное должностное лицо не совершает преступления.

9. Если на металле появились следы ржавчины, то началась коррозия. Коррозия не началась. На металле не появились следы ржавчины.

10. Если эта машина – двигатель внутреннего сгорания, то она является тепловым двигателем. Если эта машина является тепловым двигателем, то в ней топливо сжигается внутри цилиндра. Если эта машина – двигатель внутреннего сгорания, то в ней топливо сжигается внутри цилиндра.

11. Некоторый остров заселен исключительно рабами и сеньорами. На этом острове царствует некая принцесса. Сеньоры всегда говорят правду, а рабы все время лгут. Некоторые сеньоры бедны, другие богаты. Рабы также делятся на богатых и бедных. Принцесса желает выйти замуж за богатого раба. Каким же образом раб может всего лишь одним заявлением убедить принцессу в том, что он достоин быть ее супругом (какая фраза могла бы убедить принцессу в том, что говорящий эту фразу является богатым рабом)?

*ПРАКТИЧЕСКАЯ РАБОТА № 4*  
**ПРЕДСТАВЛЕНИЕ ЗНАНИЙ ПРАВИЛАМИ**  
**И ЛОГИЧЕСКИЙ ВЫВОД**

Наиболее распространённым методом представления знаний являются правила продукций, или продукционные правила. Идея этого метода, широко используемого в разработке информационных систем, принадлежит Э. Посту (1943).

Продукционная система состоит из трех основных компонентов. Первый из них – это набор правил, используемый как база знаний, поэтому его еще называют базой правил. Следующим компонентом является рабочая память (или память для кратковременного хранения), в которой хранятся предпосылки, касающиеся конкретных задач предметной области, и результаты выводов, полученных на их основании, и, наконец, следует механизм логического вывода, использующий правила в соответствии с содержимым рабочей памяти.

Основным элементом продукции является ее ядро:  $A \Rightarrow B$ . Интерпретация ядра продукции может быть различной и зависит от того, что стоит слева и справа от знака секвенции  $\Rightarrow$ . Обычное прочтение ядра продукции выглядит так: ЕСЛИ А, ТО В. Более сложные конструкции ядра допускают в правой части альтернативный выбор, например, ЕСЛИ А, ТО В1, ИНАЧЕ В2. Секвенция может истолковываться в обычном логическом смысле как знак логического следования В из истинного А (если А не является истинным выражением, то о В ничего сказать нельзя). Возможны и другие интерпретации ядра продукции, например А описывает некоторое условие, необходимое для того, чтобы можно было совершить действие В.

Пусть правило продукции имеет вид:

ЕСЛИ<условия>ТО<действие>.

Слова в угловых скобках являются посылками, или утверждениями.

Рассмотрим несложный пример. Допустим, что данные, записываемые

в рабочую память, представляют собой образцы в виде наборов символов. Например, «намерение – отдых», «место отдыха – горы» и т. п. Правила, накапливаемые в базе правил, отражают содержимое рабочей памяти. В их условной части находятся либо одиночные образцы, либо несколько условий, соединенных предлогом «и», а в заключительной части – образцы, дополнительно регистрируемые в рабочей памяти. Рассмотрим два примера подобных правил.

Правило 1.

ЕСЛИ «намерение – отдых» и «дорога ухабистая»

ТО «использовать джип»

Правило 2.

ЕСЛИ «место отдыха – горы»

ТО «дорога ухабистая»

После того, как в рабочую память записываются образцы «намерение – отдых» и «место отдыха – горы», рассматривается возможность применения этих правил. Сначала механизм вывода сопоставляет образцы из условной части правила с образцами, хранимыми в рабочей памяти. Если все образцы имеются в рабочей памяти, условная часть считается истинной, в противном случае – ложной. В данном примере образец «намерение – отдых» существует в рабочей памяти, а образец «дорога ухабистая» отсутствует, поэтому его условная часть считается ложной. Что касается правила 2, то его условная часть истинна. Поскольку в данном случае существует только одно правило с истинной условной частью, то механизм вывода сразу же выполняет его заключительную часть и образец «дорога ухабистая» заносится в рабочую память. При попытке вторично применить эти правила получается, что можно применить лишь правило 1, поскольку правило 2 уже было применено и выбыло из числа кандидатов. К этому времени содержимое рабочей памяти было дополнено новым образцом – результатом применения правила 2, поэтому условная часть правила 1 становится истинной и содержимое рабочей памяти пополняется образцом его заключительной части – «использо-

вать джип». В итоге правил, которые можно было бы применять, не остается, и система останавливается.

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 выбор для выполнения той или иной продукции из числа актуализированных. В ряде систем ИИ используют комбинации сетевых и продукционных моделей представления знаний. В таких моделях декларативные знания описываются в сетевом компоненте модели, а процедурные знания – в продукционном. В этом случае говорят о работе продукционной системы над семантической сетью.

*Классификация ядер продукций.* Ядра продукции можно классифицировать по различным основаниям. Прежде всего все ядра делятся на два больших типа: детерминированные и недетерминированные. В детерминированных ядрах при актуализации ядра и при выполнимости А правая часть ядра выполняется обязательно; в недетерминированных ядрах В может выполняться и не выполняться. Таким образом, секвенция  $\Rightarrow$  в детерминированных ядрах реализуется с необходимостью, а в недетерминированных – с возможностью. Интерпретация ядра в этом случае может, например, выглядеть так: ЕСЛИ А, ТО ВОЗМОЖНО В.

Возможность может определяться некоторыми оценками реализации ядра. Например, если задана вероятность выполнения В при актуализации А, то продукция может быть такой: ЕСЛИ А, ТО С ВЕРОЯТНОСТЬЮ Р РЕАЛИЗОВАТЬ В. Оценка реализации ядра может быть лингвистической, связанной с понятием терм – множества лингвистической переменной, например: ЕСЛИ А, ТО С БОЛЬШЕЙ ДОЛЕЙ УВЕРЕННОСТИ В. Возможны иные способы реализации ядра.

Детерминированные продукции могут быть однозначными и альтернативными. Во втором случае в правой части ядра указываются альтернативные возможности выбора, которые оцениваются специальными весами выбо-

ра. В качестве таких весов могут использоваться вероятностные оценки, лингвистические оценки, экспертные оценки и т.п.

Особым типом являются прогнозирующие продукции, в которых описываются последствия, ожидаемые при актуализации А, например: ЕСЛИ А, ТО С ВЕРОЯТНОСТЬЮ Р МОЖНО ОЖИДАТЬ В.

В настоящее время используются следующие формы продукционных правил:

ЕСЛИ<предшествующий>ТО<последующий>

ЕСЛИ<основание>ТО<гипотеза>

Таким образом, правило состоит из двух частей. Часть ЕСЛИ указывает "условия", "предшествующий" или "основание", а часть ТО представляет соответственно "заключение", "последующий" или "гипотезу". Перечисленные элементы второй части генерируются при истинности условий первой части.

Примером правила продукции может служить следующее:

ЕСЛИ клиент работает на одном месте более двух лет

ТО клиент имеет постоянную работу.

Так, в примере, представленном выше, если возраст клиента равен или больше 18 лет, то мы можем заключить, что клиент имеет право претендовать на получение кредита.

*Гипотетический силлогизм* означает, что когда заключение одного правила является посылкой другого правила, то можно установить третье правило с посылкой из первого правила и заключением из второго. Другими словами:

Условие ЕСЛИ X ТО Y

условие ЕСЛИ Y ТО Z

заключение ЕСЛИ X ТО Z

Например:

ЕСЛИ клиент женат

ТО имущество является совместной собственностью

ЕСЛИ имущество является совместным

ТО закладная на кредит может быть подписана.

Мы можем заключить, что

ЕСЛИ клиент женат

ТО закладная на кредит может быть подписана.

Модус поненс и гипотический силлогизм являются двумя правилами вывода в логике.

Рассмотрим базу знаний, написанную на примере закладной для получения кредита банка под залог недвижимого имущества (ссуды). Для получения кредита клиент должен иметь постоянную работу, приемлемый доход, хороший кредитный рейтинг и приемлемое имущество (собственность).

Если клиент не имеет постоянной работы, тогда он должен иметь адекватные активы. Величина кредита не может быть больше 80 % стоимости его имущества и клиент должен иметь 20 % ценности в кассе.

Определение "постоянная работа" состоит в том, что клиент должен иметь одну и ту же работу более двух лет. Определение "адекватные активы" заключается в том, что имущество клиента должно быть оценено как десятикратная величина кредита или клиент должен иметь ликвидные активы, оцениваемые как пятикратная величина кредита. "Приемлемое имущество" – это имущество, находящееся на территории расположения банка.

Проверка кредитного рейтинга проводится вручную.

Смысл "адекватного дохода" состоит в следующем. Если клиент *одиночка*, то оплата закладной должна быть менее чем 70 % его чистого дохода.

Если клиент женат, то оплата закладной должна быть менее чем 60 % чистого дохода семьи.

В продукционных правилах мы будем использовать соединители И, ИЛИ и НЕ.

1. ЕСЛИ клиент имеет постоянную работу

И клиент имеет адекватный доход

И имущество приемлемо

И клиент имеет хороший кредитный рейтинг

И величина кредита меньше 80% стоимости имущества

И клиент имеет 20% стоимости имущества в кассе

ТО одобрить кредит.

2. ЕСЛИ клиент имеет адекватные активы

И клиент имеет адекватный доход

И имущество приемлемо

И клиент имеет хороший кредитный рейтинг

И величина кредита меньше 80% стоимости имущества

И клиент имеет 20% стоимости имущества в кассе

ТО одобрить кредит.

3. ЕСЛИ клиент имеет работу

И клиент имел более чем в течении двух лет эту работу

ТО клиент имеет постоянную работу.

4. ЕСЛИ имущество находится в зоне расположения банка

ИЛИ имущество в запрещенном списке

ТО имущество приемлемо.

5. ЕСЛИ доход имеется и адекватен

ИЛИ доход одиночки адекватен

ТО клиент имеет адекватный доход.

6. ЕСЛИ клиент женат

И оплата закладной меньше 60% чистого дохода семьи

ТО доход семьи адекватен.

7. ЕСЛИ клиент не женат

И оплата закладной меньше 70% чистого дохода клиента

ТО доход одиночки адекватен.

8. ЕСЛИ стоимость имущества клиента больше десятикратной величины

кредита

ИЛИ ликвидные активы клиента больше пятикратной величины кре-

дита

ТО клиент имеет адекватные активы.

Популярность продукционных правил объясняется несколькими причинами:

1. Большинство человеческих знаний можно представить в виде правил продукций.
2. Модульность продукции позволяет добавлять в систему новые продукции без изменения прежних.
3. Продукции могут реализовать любые алгоритмы и, следовательно, любые процедурные знания.
4. Параллелизм и асинхронность продукций делают их удобной моделью вычислений, отвечающей подобным требованиям новых поколений ЭВМ.

Недостаток продукции состоит в том, что при их большом количестве становится трудоемкой проверка непротиворечивости системы продукций, – например, при добавлении новых правил.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что является фактом в продукционном правиле?
2. Какая часть правила называется посылкой, какая выводом?
3. Что означает гипотетический силлогизм?
4. Чем объясняется популярность продукционных правил?
5. В чем состоит недостаток продукций?
6. Какова классификация ядер продукции?

### ЗАДАНИЕ

Представить знания о заданной предметной области в виде набора правил (не менее 20).

1. Диагностика неисправностей персонального компьютера. Экспертная система должна исследовать ситуацию и попытаться определить на общем уровне, допускает ли ошибки пользователь или, действительно, имеется не-

исправность в системном блоке, на диске, в мониторе и т.д. Возможный путь проектирования – беседа с мастером-профессионалом. При оценке ситуации подразумеваются грубые функциональные тесты, без глубокого анализа электронных элементов.

2. Подбор субоптимальной конфигурации персонального компьютера с учетом субъективных и объективных потребностей заказчика.

3. Подбор субоптимальной конфигурации локальной компьютерной сети с учетом множества эксплуатационных, финансовых и прочих важных критериев.

4. Диагностика широко распространенных заболеваний человека по совокупности симптомов. Диагностируется не менее 20 болезней с учетом 15 типовых симптомов. Каждый симптом может указывать на несколько болезней (возможно, с разной степенью уверенности).

5. Консультация в отношении покупки автомобиля с учетом субъективных факторов, объективных потребностей и платежеспособности клиента, сезона и др.

6. Консультация в отношении покупки недвижимости с учетом связанных с этим важных факторов (надежность продавца, платежеспособность покупателя, страхование сделки, изменение цен и банковских процентных ставок и др.).

7. Диагностика причины зависания в процессе загрузки операционной системы ПЭВМ.

8. Подбор членов экипажа с учетом их психологической совместимости и возраста. При этом учитываются индивидуальные особенности претендентов, пожелания тех или иных кандидатов работать вместе, а также их возраст (командир должен быть старше всех остальных членов экипажа).

9. Информация об авиалиниях, связывающих Москву с некоторыми городами РФ. С какими городами Москва связана прямыми беспосадочными авиалиниями: с указанием всех городов, из которых имеются обратные прямые беспосадочные рейсы.

10. Консультация в отношении приема лекарств при болезнях родственных вирусной инфекции или гриппу. В зависимости от конкретных симптомов заболевания решить, обращаться ли к врачу.

#### *ПРАКТИЧЕСКАЯ РАБОТА № 5.*

#### *СЕМАНТИЧЕСКИЕ СЕТИ*

Разработка семантических сетей относится к 1960 г., когда они использовались для моделирования обработки естественного языка.

"Семантика" определяется как значение, смысл слова, переданные с помощью каких-либо представлений и выражений. Другими словами, семантика означает определенные отношения между символами и объектами, представленными этими символами.

Семантическая сеть [6] – структура для представления знаний в виде узлов, соединенных дугами. Самые первые семантические сети были разработаны в качестве языка-посредника для систем машинного перевода, а многие современные версии по своим характеристикам до сих пор сходны с естественным языком. Однако последние версии семантических сетей стали более мощными и гибкими, они составляют конкуренцию фреймовым системам, логическому программированию и другим языкам представления.

С конца 50 гг. были созданы и использованы на практике десятки вариантов семантических сетей. Несмотря на то, что терминология и их структура различаются, существуют элементы сходства, присущие практически всем семантическим сетям:

- 1) узлы семантических сетей представляют собой концепты предметов, событий, состояний;
- 2) различные узлы одного концепта относятся к различным значениям, если не помечено, что они относятся к одному концепту;
- 3) дуги семантических сетей создают отношения между узлами-концептами (пометки над дугами указывают на тип отношения);

4) некоторые отношения между концептами представляют собой лингвистические падежи, – такие как агент, объект, реципиент и инструмент (другие означают временные, пространственные, логические отношения и отношения между отдельными предложениями);

5) концепты организованы по уровням в соответствии со степенью обобщенности – как, например, сущность, живое существо, животное, плотоядное.

Однако имеются и различия: понятие значения с точки зрения философии; методы представления кванторов общности и существования в логических операторах; способы манипулирования сетями и правила вывода, терминология. Все это у различных авторов варьируется. Несмотря на некоторые различия, сети удобны для чтения и обработки компьютером, а также достаточно мощны, чтобы представить семантику естественного языка.

Семантические сети предназначены для семантического моделирования реальной действительности с возможностью представления как экстенциональной, так и интенциональной информации. Семантическую сеть можно рассматривать как маркированный ориентированный граф с помеченными узлами и дугами. Узлам соответствуют некоторые объекты, а дугам – семантические отношения между этими объектами. Метки, приписываемые узлам, выделяют множество рассматриваемых объектов и выступают в качестве их имени. В роли таких имен могут быть слова естественного языка. Метки, приписываемые дугам, соответствуют элементам множества отношений, заданных на графе.

В семантической сети могут быть выделены подграфы определенной структуры, называемые *высказываниями*. Каждый такой подграф представляет собой граф, корнем которого является предикатный узел, остальные узлы называются концептуальными. Высказывание является минимальной единицей информации, вводимой и хранящейся в семантической сети. Разделение узлов на предикатные и концептуальные возможно лишь в пределах одного высказывания. В структуре семантической сети один и тот же узел может

быть предикатным относительно одного высказывания и концептуальным относительно другого.

В качестве структурной модели долговременной памяти психолог Куиллиан предложил модель понимания смысла слов, получившую название TLC-модель (доступный механизм понимания языка). Для описания структуры долговременной памяти в ней была использована сетевая структура как способ представления семантических отношений между концептами (словами).

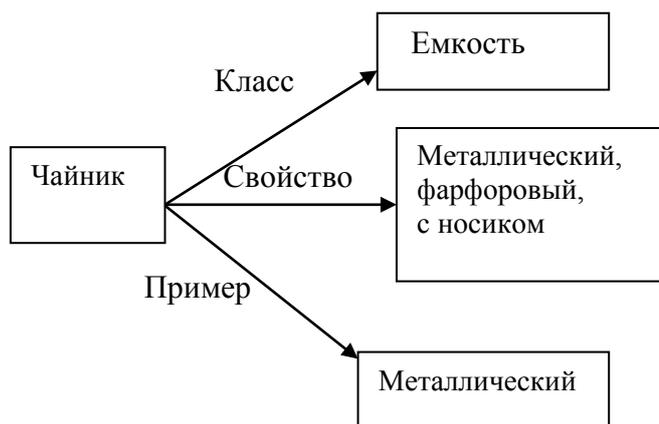


Рис. 1. Образец семантической сети.

В модели Куиллиана концептуальные объекты представлены ассоциативными сетями, состоящими из вершин, показывающих концепты, и дуг, показывающих отношения между концептами. Подобная ассоциативная структура называется плоскостью, описываемые концепты объекта – называются вершинами типа, а связанные с ними соответствующие отдельные слова – вершинами лексем. В любой плоскости существует одна вершина типа и только необходимое для определения концептов, описывающих его, число вершин лексем. Вершины лексем определяют всевозможные сущности, имеющие место в реальном мире.

В семантических сетях используются четыре основных типа объектов: понятия, события, свойства и значения.

*Понятия* представляют собой сведения об абстрактных или физических объектах предметной области. Они могут быть заданы множеством доменов (параметров или констант).

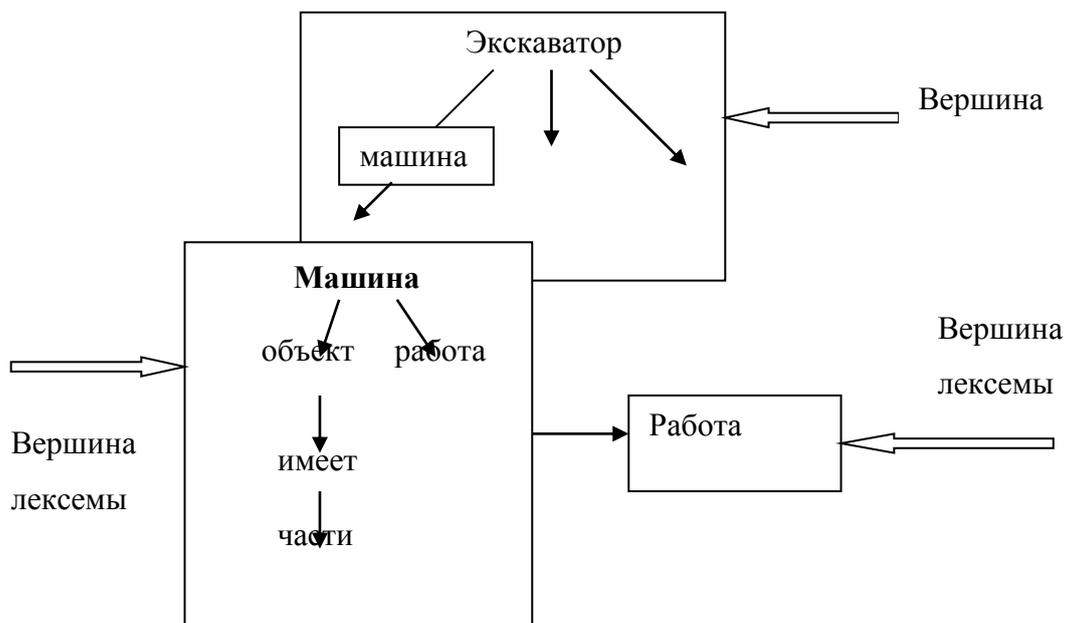


Рис.2. Представление знаний в долговременной памяти для TLC-моделей Куиллиана.

*События* – это действия, способные перевести предметную область из текущего состояния в некоторое новое. Можно перевести предметную область в определенное желаемое (целевое) состояние, поставить задачу, отыскать на семантической сети путь событий, приводящих к целевому состоянию.

*Свойства* используются для характеристики (уточнения) понятий, событий, процессов и т.д. Свойствами понятий могут быть цвет, размеры, качество, а свойствами действий – время, продолжительность, место и т.д.

*Значения* – это значения из какого-нибудь множества, которые могут принимать свойства.

Рассмотрим пример семантической сети (рис. 3), которая соответствует действию "Иванов руководит Петровым". Она представляется одним событием РУКОВОДИТ и двумя вершинами понятий ИВАНОВ и ПЕТРОВ. Дуги, соединяющие вершины-события и вершины-понятия, указывают роли понятий в событиях. Так, в событии РУКОВОДИТ понятие ИВАНОВ играет роль агента, а ПЕТРОВ – объекта воздействия. Свойством понятия ИВАНОВ является ДОЛЖНОСТЬ со знанием ПРОФЕССОР.

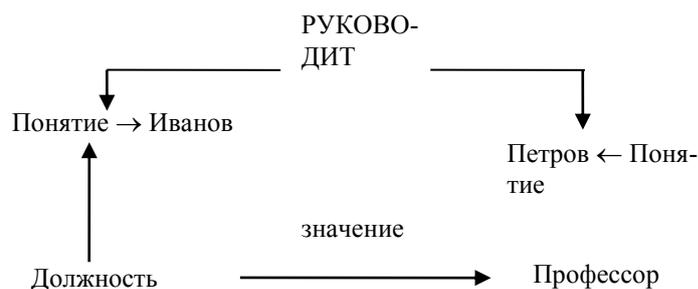


Рис. 3. Пример семантической сети

Основными связями для семантических сетей, с помощью которых формируются понятия, являются:

структурные – класс, к которому принадлежит данное понятие;

отличительные свойства, выделяющие понятие из всех прочих понятий этого класса;

примеры данного понятия;

связи типа "часть – целое" ("класс – подкласс", "элемент – множество" и т.п.);

функциональные связи (определяемые обычно глаголами "выполняет", "влияет"…);

количественные ("больше", "меньше", "равно");

пространственные ("близко", "далеко", "за", "перед", "внутри");

временные ("раньше", "позже", "во время");

атрибутивные ("имеет свойство", "имеет значение");

логические (И, ИЛИ, НЕ).

На самой СС принадлежность элемента к некоторому классу или части к целому передается с помощью связок "это есть" и "часть от" соответственно. Свойства описываются связками "есть" и "имеет". Отбор понятий и связок определяется предметной областью и назначением ССИ.

В качестве примера может быть приведена часть семантической сети, относящейся к понятию "фрукты".

Существует несколько классификаций семантических сетей, связанных с типами отношений между понятиями.

По количеству типов отношений:

однородные (с единственным типом отношений);

неоднородные (с различными типами отношений).

По типам отношений:

бинарные (отношения связывают два объекта).

N-арные (в которых есть специальные отношения, связывающие более двух понятий).

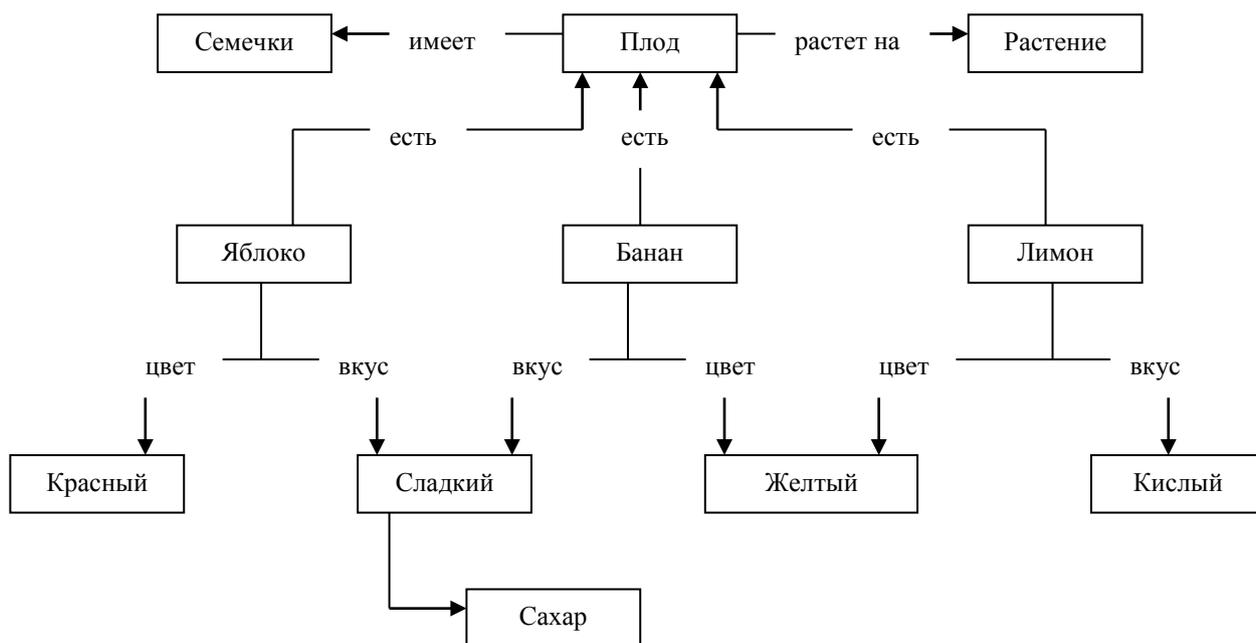


Рис. 4. Фрагмент семантической сети «Фрукты».

### Правила построения семантических сетей.

Семантические сети получаются из концептуальных графов с помощью специальных правил.

*Правило конъюнкции.* Рассмотрим набор из трех фраз:

1 – я: Завод “Интеграл” производит Схему\_14;

2 – я: “Интеграл” поставляет Схему\_14 фирме “Луч”;

3 – я: Фирма “Луч” использует Схему\_14 для сборки часов.

Каждую из этих фраз можно представить концептуальным графом. Используя правило конъюнкции, можно получить один концептуальный граф  $g$  из двух графов  $g1$  и  $g2$ .

Это правило состоит в следующем:

если узлы-концепты  $c1$  и  $c2$  соответственно в графах  $g1$  и  $g2$  идентичны, то граф  $g$  получается удалением  $c2$  и соединением с  $c1$  всех связывающих узлов, которые были связаны с  $c2$  и  $g2$ .

Применим правило конъюнкции к приведенным выше трем фразам и получим семантическую сеть (рис. 5).

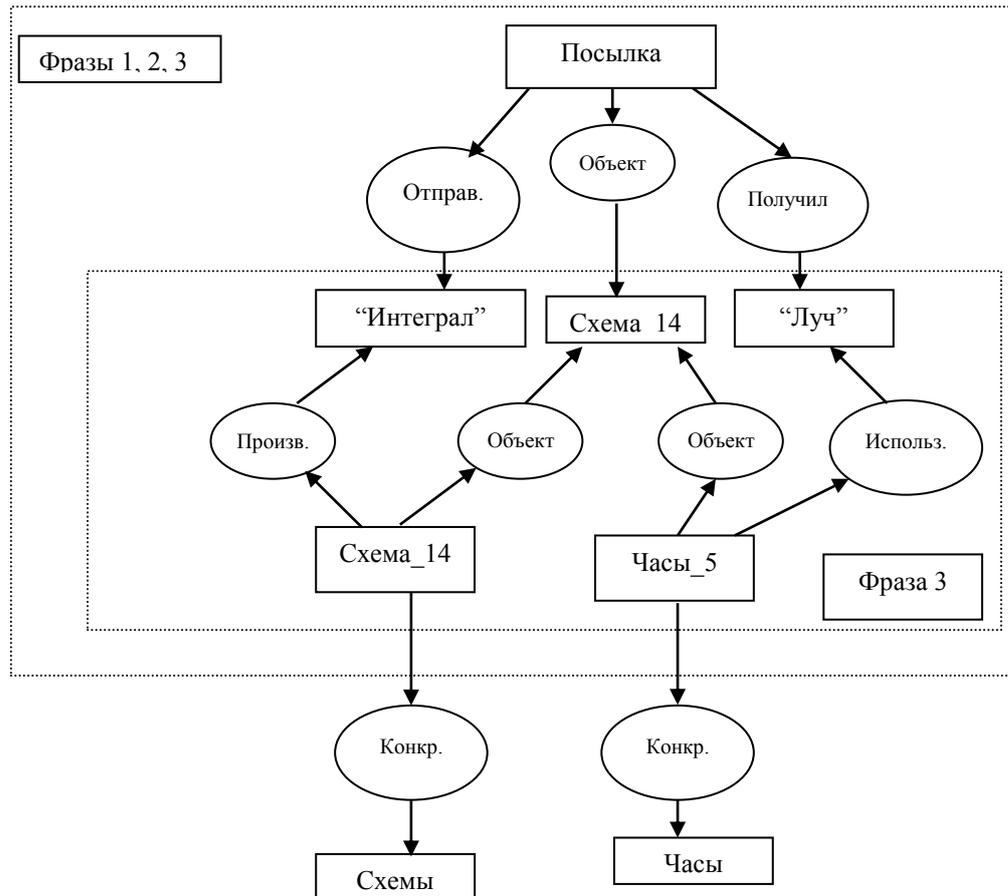


Рис.5. Концептуальный граф и контекстное представление

*Правило упрощения.* Если концептуальный граф  $g$  содержит два идентичных (соединенных с одними и теми же узлами-концептами) связывающих узла, то можно удалить один из них со связанными с ним стрелками.

*Правило копирования.* Граф  $g$  есть копия графа  $g1$ .

*Использование типов.* Отношение принадлежности типу представляется связывающим узлом КОНКР (от слова "конкретизация"). Так, "Схема\_14" относится к типу "схема", что указано узлом КОНКР. Отношение принадлежности множеству представляется связывающим узлом ЭЛЕМ (от

слова "элемент"). Так, завод "Интеграл" принадлежит множеству заводов электроники ("электрозавод").

Отношение принадлежности подмножества некоторому множеству представляется узлом ПОДМН.

Связь между первым типом и более общим вторым типом осуществляется связующим узлом ЭТО.

*Представление «совокупность – ссылка».* Семантические сети образуются последовательностью трех узлов, соединенных так: за узлом, представляющим конкретное значение (Схема\_14), следует связывающий узел КОНКР, сопровождаемый узлом-совокупностью (схема). Эти три узла можно сгруппировать и изобразить одним узлом-прямоугольником, состоящим из двух полей: из «поля-совокупности», содержащего некоторый тип, и следующего за ним «поля ссылки», конкретизирующего тип из первого поля. Например,



В поле конкретизации может находиться переменная. Так, [схема: x] означает просто какой-то объект типа «схема». Отметим, что переменную можно просто опустить, условившись, что представление [схема] эквивалентно [схема: x].

*Канонические графы.* Синтаксически правильно построенные концептуальные графы могут семантически быть некорректными, как собственно и предложения, которым они соответствуют, т. е. некоторые комбинации узлов бессмысленны. Чтобы исключить графы нереальных (невозможных) ситуаций предметной области, определяют так называемые канонические (семантически корректные) графы разрешенных комбинаций слов. Следовательно, знания представляются каноническими графами исходя из личного опыта. Новые, канонические графы строятся из уже имеющихся по правилам: конъюнкции, упрощения, копирования и ограничения.

Правила конъюнкции были рассмотрены выше.

Правило ограничения заключается в следующем. Тип любого концепта

с концептуального графа  $g$  можно записать неким подтипом, т.е. конкретизировать концепт  $c$ . Проиллюстрируем это на примере концептуального графа, представленного на рис. 6.

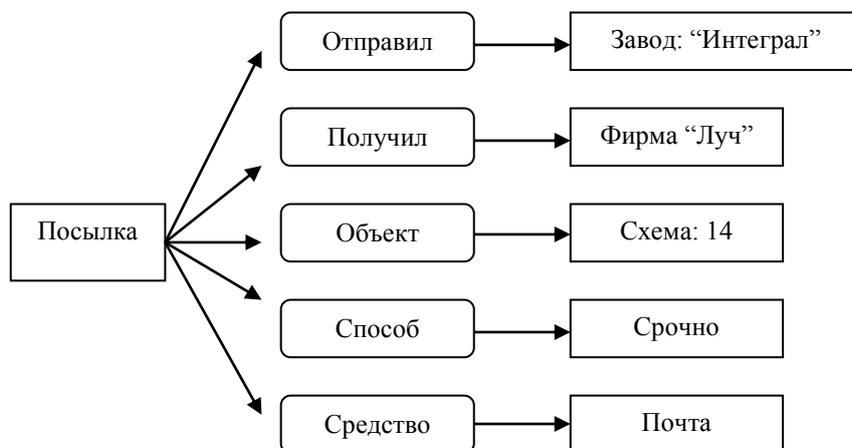


Рис. 6. Пример построения семантической сети.

Граф без элементов  $s(2)$  интерпретируется фразой ~Интеграл срочно поставляет схему кому-то ~. Граф без элементов  $s(1)$  означает ~Интеграл поставляет схему фирме “Луч” почтой~. Применения правила конъюнкции к этим двум графам дает весь граф на рис.6. Можно было бы применить к последнему графу правило ограничения, заменяя обозначенную  $X$  совокупность ~кому-то~ меньшей совокупностью ~часовые заводы~. Правило ограничения сужает концепты, а конъюнкции – добавляет условия на граф. На рис.7. представлен пример семантической сети для моделирования закладной под кредит.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. На чем основано представление знаний с помощью семантической сети?
2. Из чего состоит семантическая сеть?
3. Чем отличается семантическая сеть от других моделей представления знаний?
4. Какие типы отношений используются в семантических сетях?
5. Как классифицируются семантические сети?

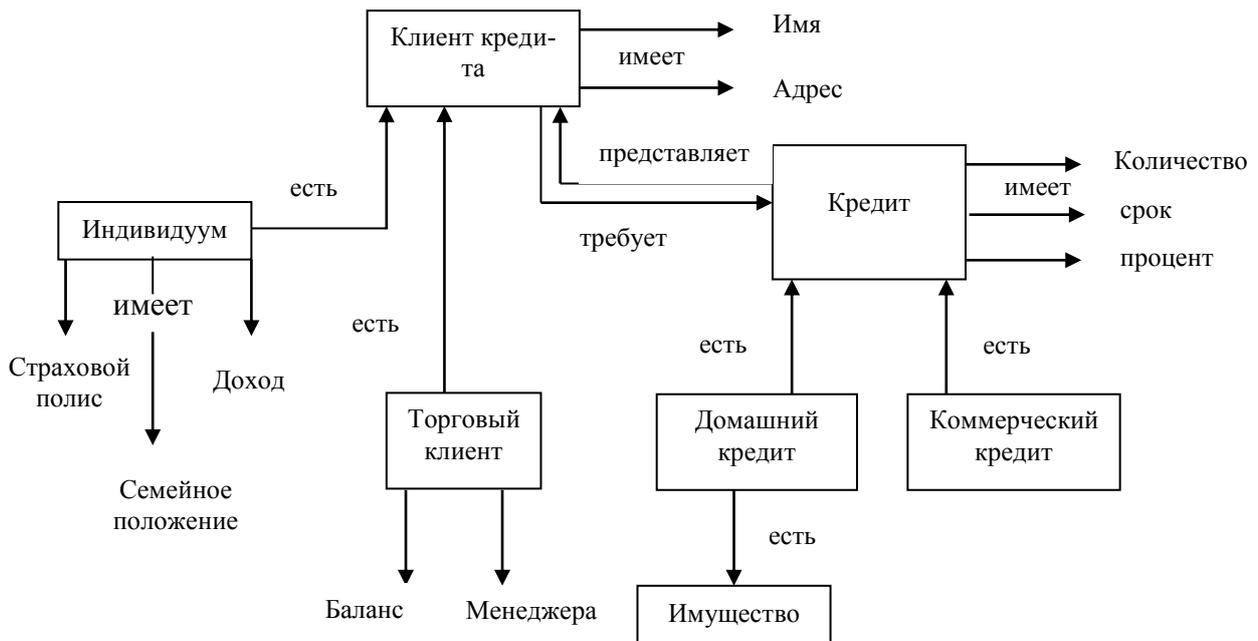


Рис. 7. Семантическая сеть для моделирования закладной под кредит.

## ЗАДАНИЕ

Представить знания о заданной предметной области в виде семантической сети (по вариантам заданий, приведенных в практической работе № 4). Привести графическую интерпретацию.

### ПРАКТИЧЕСКАЯ РАБОТА № 6

#### **ПРЕДСТАВЛЕНИЕ ЗНАНИЙ ФРЕЙМАМИ И ВЫВОДЫ**

Теорию фреймов опубликовал в 1975 г. М. Минский. Она относится к психологическим понятиям и касается способов понимания того, как мы воспринимаем (видим, слышим) явления, процессы, объекты и т.п.

В основе теории фреймов лежит восприятие фактов посредством полученной извне информации о некотором явлении с уже имеющимися данными, накопленными опытным путем или полученными в результате вычислений. Когда человек попадает в новую ситуацию, он вызывает из своей памяти основную структуру, называемую фреймом. *Фрейм* (рамка) – это единица представления знаний, заложенных в прошлом, детали которой могут

быть изменены согласно текущей ситуации. Класс некоторых объектов (процессов) может определяться одним типичным (базовым) объектом, включающим наиболее существенные характеристики объектов данного класса. Так, некоторую характеристику объекта можно представить тройкой (Объект, атрибут<sub>j</sub> значение<sub>j</sub>).

Собрав все тройки, касающиеся данного объекта, получим объектное представление области рассуждений относительно данного объекта. Общая форма этого представления такова:

$$\text{Объект(атрибут}_i \text{ значение}_i), j= 1, \dots, m$$

Таким образом, вместо построения различных независимых формул строим более крупную структуру полной информации об объекте, которую называют фреймом. Если требуется информация о некотором объекте, то обращаются к соответствующему фрейму, внутри которого находятся свойства и факты относительно рассматриваемого объекта.

Фрейм содержит как информационные, так и процедурные элементы, которые обеспечивают преобразование информации внутри фрейма и связи его с другими фреймами. Элементами фрейма являются так называемые слоты, которые могут быть пустыми и заполняться в процессе активизации фрейма в соответствии с определенными условиями. Таким образом, фреймы представляют собой декларативно-процедурные структуры, т.е. совокупность описаний и (возможно) связанных с ними процедур, доступ к которым выполняется прямо из фрейма.

### **Структура данных фрейма**

Фрейм можно представить в виде таблицы (отношения), строки которой соответствуют слотам фрейма, а столбцы – атрибутам (характеристикам) объекта. Имя таблицы является именем фрейма, и оно уникально. Каждый слот содержит следующие атрибуты: имя слота наследования, указатель наследования, тип данных, значение слота, демон.

ТИП ДАННЫХ определяет, что слот либо имеет численное значение, либо является именем другого фрейма. К типам данных относится: FRAME

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

ЗНАЧЕНИЕ СЛОТА соответствует указанному типу данных этого слота; кроме того, должно выполняться условие наследования. Значениями слотов могут быть имена других фреймов, что обеспечивает связь (вложенность) между фреймами.

ДЕМОН – автоматически запускаемая процедура при обращении к слоту и выполнении некоторого условия. Например, демон типа IF-NEEDED запускается, если в момент обращения к слоту его значение не было установлено; IF-ADDED запускается при подстановке в слот значения, IF-REMOVED – при стирании значения слота.

Присоединенная процедура может использоваться в качестве значения слота и запускаться по сообщению, переданному из другого фрейма. Когда мы говорим, что фреймы, как модели представления знаний, объединяют декларативные и процедурные знания, то считаем демоны и присоединенные процедуры процедурными знаниями.

Фрейм состоит из произвольного числа слотов, среди которых имеются системные слоты и слоты, определяемые пользователем. Каждый слот характеризуется определенной структурой и уникальным именем внутри данного фрейма. В качестве системных могут быть, например, определены следующие слоты: указатель фрейма-родителя, указатель прямого дочернего фрейма, пользователь фрейма, дата определения фрейма и его последней модификации, а также некоторые другие. Системные слоты используются при редактировании баз данных и управлении выводом.

Понятие «наследование свойств» позволяет фреймам заимствовать свойства, которые имеют другие фреймы. УКАЗАТЕЛИ НАСЛЕДОВАНИЯ касаются только фреймовых систем иерархического типа, основанных 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:

*U* (unique – уникальный); *S* (same – такой же); *R* (range – установление границ); *O* (override – переопределить) и др. Указатель наследования *U* показывает, что каждый фрейм должен иметь слоты с различными значениями; *S* – все слоты имеют одно значение; *R* – значения слотов фрейма нижнего уровня должны находиться в пределах указанных значений слотов фрейма верхнего уровня; *O* – при отсутствии указания значение слота фрейма верхнего уровня становится значением слота фрейма нижнего уровня.

### **Свойства фреймов**

Рассмотрим основные свойства фреймов.

*1. Базовый тип.* Базовые фреймы используются для указания наиболее важных объектов, позволяют добиться быстрого понимания сущности данного предмета. На основании базовых фреймов строятся фреймы для новых состояний. При этом каждый фрейм содержит слот-указатель подструктуры, что позволяет различным фреймам совместно использовать одинаковые части.

*2. Процесс сопоставления.* Во фреймовой системе осуществляется поиск фрейма, который соответствует (релевантен) цели (данной ситуации). Другими словами, сопоставляются значения (ограничения) слота фрейма во фреймовой системе со значениями атрибутов цели. Процесс сопоставления осуществляется следующим образом:

а) сначала с помощью предложения и интуиции выбирается некоторый базовый фрейм с учетом выявленных особенностей, релевантности, т.е. посредством подфреймов. Данный фрейм подтверждает или не подтверждает свою релевантность. При этом в соответствии с текущей целью определяется, какое ограничение слота следует использовать при сопоставлении. Если фрейм подходит, то процесс сопоставления завершается, в противном случае выполняется пункт б);

б) если в данном фрейме имеется слот, значение которого отрицательно влияет на сопоставление, то нужно присвоить слоту надлежащее значение;

в) если два предыдущих шага не дают результата, то управление передается другому надлежащему фрейму этой или другой фреймовой системы. Если последнее сопоставление заканчивается безрезультатно, то задача не имеет решения.

*3. Иерархическая структура.* Фрейм обычно соответствует некоторой иерархической структуре, особенность которой состоит в том, что значения атрибутов фрейма верхнего уровня совместно используются всеми фреймами нижних уровней, связанных с верхними (рис. 8). Такая структура позволяет удобно систематизировать и записать схожие понятия, добавлять новые понятия или знания в соответствующие позиции иерархии, упрощает обнаружение противоречий в знаниях, просмотр знаний и делает фреймовую систему более гибкой.

Пример 1. Рассмотрим пример фрейма в системе моделирования процесса выдачи банковского кредита под залог (рис.8). Эта система содержит фрейм «заемщика», который представляет собой класс клиентов и принадлежит к суперклассу заемщиков кредита. Он содержит характеристики, относящиеся к человеку-клиенту, такие как цель использования кредита, информацию об обеспечении клиента, его ответственности, имуществе и т. д.

Наследование свойств у фреймов представляется отношениями каждого фрейма с другими фреймами. Например, фрейм "индивидуальный клиент" имеет в качестве его суперкласса фрейм "клиент" и наследует его характеристики.

Важным является то, что фреймы имеют сложную структуру данных, в которой атрибуты фрейма сами являются фреймами. Например, "индивидуальный клиент" для фрейма "кредит под залог" сам является фреймом с присущими ему атрибутами.

Процедурный характер фрейма "кредит под залог" состоит в том, что значение слота "плата залога" может быть результатом вычисления по формуле, в которой используются слоты "количество", "процентная ставка" и "продолжительность".



Рис.8. Пример фрейма

В фреймовых системах отсутствует специальный механизм управления выводом, поэтому пользователь может реализовать данный механизм с помощью присоединенной процедуры. В то же время это дополнительная нагрузка для пользователя, поэтому язык представления знаний фреймами ориентирован на специалистов по искусственному интеллекту.

Основной причиной того, что фреймовые системы являются наиболее благоприятной средой для исследований по искусственному интеллекту, сле-

дует считать то, что они в целом соответствуют многочисленным требованиям, касающимся представления знаний:

1. Для систематизированного управления сложными знаниями большого объема желательно организовать эти знания на основе концептуальных объектов.

2. В целях увеличения гибкости системы следует сделать возможным представление в виде комбинации декларативных и процедурных знаний.

3. Объекты представляют собой иерархическую структуру.

Существуют три способа управления выводом во фреймовой системе:

1. Способ, при котором фреймовая система используется как база данных, основанная на фактах.

2. Когда во фреймовой системе ограничена роль присоединенной процедуры. Фреймовая система используется внешней базой правил и механизмом ввода и отвечает на сообщения. Этот способ применяется мощными продукционными системами с базой данных фреймового типа, снабженной механизмом наследования и демона.

3. Способ с обменом сообщениями. Основан на постепенном продвижении к цели посредством поочередной передачи сообщений между фреймами. Однако способ требует тщательного проектирования всей системы.

Параллельно с языками фреймов существуют объектно-ориентированные программные языки, используемые для составления программ, но имеющие некоторые свойства языков фреймов (использование слотов для детальной, доскональной классификации объектов и пр.). Отличие их от языков фреймов в том, что фреймовые языки направлены на более обобщенное представление информации об объекте.

Одна из трудностей представления знаний и языка фреймов – является отсутствие формальной семантики. Это затрудняет сравнение свойств представления знаний различных языков фреймов, а также полное логическое объяснение языка фреймов.

Пример 2. Знания о предметной области представлены в виде фрейм-модовой структуры.

Фрейм "Компьютер" – фрейм класса

Слот наследования	Указатель наследования	Тип данных	Значение слота	Демон
Слот наследования	Уникальные атрибуты	Родительский фрейм	Встречаются впервые	Родителей не имеет, является базовым фреймом
Описание	Уникальные атрибуты	Текст	Встречаются впервые	Компьютер – класс ПЭВМ
Свойства компьютера	Уникальные атрибуты	Текст	Встречаются впервые	Имеет монитор, клавиатуру, мышь, МП, материнскую плату, элементы памяти, винчестер

Фрейм "Работа в Internet" – фрейм шаблон

Слот наследования	Указатель наследования	Тип данных	Значение слота	Демон
Слот наследования	Уникальные атрибуты	Указатель на родительский фрейм	Унаследовано от фрейма "Компьютер"	Родитель – фрейм компьютер
Описание	Уникальные атрибуты	Текст	Унаследовано от фрейма "Компьютер"	Компьютер – класс ПЭВМ
Свойства компьютера	Унаследованные атрибуты	Текст	Унаследовано от фрейма "Компьютер"	Имеет монитор, клавиатуру, мышь, МП, материнскую плату, элементы памяти, винчестер.
Свойства мультимедиа-компьютера	Унаследованные атрибуты	Указатель на фрейм "Мультимедиа-компьютер"	Унаследовано от фрейма "Мультимедиа-компьютер"	Имеет звуковую карту, колонки, CD-ROM.
Свойства работы в Internet	Уникальные атрибуты	Текст	Встречаются впервые	Имеет модем

## Фрейм "Принтер" – фрейм шаблон

Слот наследования	Указатель наследования	Тип данных	Значение слота	Демон
Слот наследования	Уникальные атрибуты	Указатель на родительский фрейм	Унаследовано от фрейма "Компьютер"	Родитель – фрейм компьютер
Описание	Уникальные атрибуты	Текст	Унаследовано от фрейма "Компьютер"	Компьютер – класс ПЭВМ
Свойства компьютера	Унаследованные атрибуты	Текст	Унаследовано от фрейма "Компьютер"	Имеет монитор, клавиатуру, мышь, МП, материнскую плату, элементы памяти, винчестер.
Свойства графической станции	Унаследованные атрибуты	Указатель на фрейм "Графическая станция"	Унаследовано от фрейма "Графическая станция"	Имеет 3D-видеокарту, световое перо, записывающий CD-ROM
Свойства принтера	Уникальные атрибуты	Текст	Встречаются впервые	Имеет принтер

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое фрейм?
2. Какие основные свойства фреймов вы знаете?
3. Какова структура данных фрейма?
4. Что такое «наследование» свойств?
5. Как осуществляется механизм вывода во фрейме?
6. Какую роль играет иерархическая структура во фрейме?

### ЗАДАНИЕ

Представить знания о заданной предметной области в виде фреймовой структуры (по вариантам заданий приведенных в практической работе № 4).

## Литература

1. Станкевич, Л. А. Интеллектуальные системы и технологии : учебник и практикум для вузов / Л. А. Станкевич. — Москва : Издательство Юрайт, 2020. — 397 с. — (Высшее образование). — ISBN 978-5-534-02126-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <http://www.biblio-online.ru/bcode/450773>
2. Бессмертный, И. А. Интеллектуальные системы : учебник и практикум для вузов / И. А. Бессмертный, А. Б. Нугуманова, А. В. Платонов. — Москва : Издательство Юрайт, 2020. — 243 с. — (Высшее образование). — ISBN 978-5-534-01042-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <http://www.biblio-online.ru/bcode/451101>
3. Павлов С.Н. Системы искусственного интеллекта. Часть 2 [Электронный ресурс] : учебное пособие / С.Н. Павлов. — Электрон. текстовые данные. — Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2011. — 194 с. — 978-5-4332-0014-2. — Режим доступа: <http://www.iprbookshop.ru/13975.html>
4. Павлов С.Н. Системы искусственного интеллекта. Часть 1 [Электронный ресурс] : учебное пособие / С.Н. Павлов. — Электрон. текстовые данные. — Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2011. — 176 с. — 978-5-4332-0013-5. — Режим доступа: <http://www.iprbookshop.ru/13974.html>
5. Интеллектуальные системы [Электронный ресурс] : учеб. пособие / А. М. Семенов [и др.]. - Оренбург : Оренбург. гос. ун-т, 2013. - 236 с. <http://www.iprbookshop.ru/30055>
6. Учебно-методическое пособие по дисциплине Интеллектуальные информационные системы и технологии [Электронный ресурс] / . — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2014. — 24 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/61479.html>
7. Интеллектуальные системы и технологии [Электронный ресурс] : сб. учеб.-метод. материалов для направления подготовки 09.03.02 "Информационные системы и технологии" / АмГУ, ФМиИ ; сост. И. М. Акилова. - Благовещенск : Изд-во Амур.гос. ун-та, 2017. - 60 с. - Б. ц. [http://irbis.amursu.ru/DigitalLibrary/AmurSU\\_Edition/10310.pdf](http://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/10310.pdf)
9. Основы логического программирования с использованием языка Пролог [Электронный ресурс] : лаб. практикум / И. М. Акилова, Н. В. Назаренко ; АмГУ, ФМиИ. - Благовещенск : Изд-во Амур.гос. ун-та, 2011. - 116 с. - Б. ц. [http://irbis.amursu.ru/DigitalLibrary/AmurSU\\_Edition/3592.pdf](http://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/3592.pdf)