ЛАБОРАТОРНАЯ РАБОТА №4

Представление и обработка знаний с использованием логических функций

Цель работы. Изучение возможностей представления и обработки знаний в экспертных системах с помощью логических функций

Представление знаний с помощью логических функций

Во многих случаях знания человека, выраженные в форме высказываний, могут быть представлены с помощью функций логических переменных (булевых функций). Напомним, что логическая (булева) переменная – это переменная, принимающая только два значения: “истина” (1) или “ложь” (0).
Такие переменные соответствуют высказываниям, описывающим знания человека. Из всех известных функций логических переменных, для представления знаний в форме высказываний обычно достаточно использовать функции отрицания, дизъюнкции (логическое “или”), конъюнкции (логическое “и”), импликации. Отрицание логической переменной ~A (“не A”) принимает значение “истина”, если переменная A имеет значение “ложь” (высказывание A ложно). Таким образом, A=1, если ~A=0, и наоборот.

Дизъюнкция Av B (“A или B”) истинна, если *хотя бы одна* из переменных имеет значение “истина” (хотя бы одно из высказываний A или B истинно).

Конъюнкция A&B (“A и B”) истинна, если *обе* переменные имеют значения “истина” (оба высказывания A и B истинны).

Импликация A→B (“A влечет B”, “из A следует B”) *ложна*, если переменная A имеет значение “истина”, а B – “ложь” (высказывание A истинно, а B – ложно).

В таблице 5.1 приведены значения функций дизъюнкции, конъюнкции и импликации при различных значениях аргументов. Такая таблица называется таблицей истинности.

Таблица 5.1

A	**B**	**Av B**	**A&B**	**A→B**
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	1

В некоторых случаях для упрощения выражений с логическими функциями удобно использовать следующие равенства:

~(B & A) =~ B v ~A
~(B v A) = ~ B & ~A
B → A = ~B v A

Эти равенства легко проверить, составив таблицу истинности для левой и правой части каждого из них.

Пример. Пусть при некотором заболевании могут использоваться лекарства A, B и C. Имеются следующие сведения:

• если используется лекарство А и не используется В, то необходимо использовать лекарство С;
• если используется лекарство С, то необходимо использовать хотя бы одно из лекарств А или В;
• лекарства А и С несовместимы.
Введем логические переменные А, В и С. Пусть эти переменные принимают значение “истина”, если соответствующее лекарство используется, и “ложь” – если не используется. Приведенные выше указания об использовании лекарств можно записать с помощью логических функций следующим образом:

$(A \& \sim B \rightarrow C)$
$C \rightarrow A \vee B$
$A \rightarrow \sim C$

Представление логических функций в алгебраической форме

Для удобства компьютерной обработки во многих случаях желательно перейти от представления знаний с помощью логических функций к их записи в алгебраической форме, т.е. с помощью обычных переменных, над которыми можно выполнять операции сложения, умножения, сравнения и т.д. Основные правила замены логических функций на алгебраические выражения приведены
в табл.5.2.

Таблица 5.2

Логическая функция	Алгебраическое выражение
Отрицание $\sim A$	$1-A$
Дизъюнкция $(A \vee B)$	$A+B-A \cdot B$
Конъюнкция $(A\&B)$	$A \cdot B$
Импликация $(A \rightarrow B)$	$B-A \geq 0$

Покажем возможность замены логической операции отрицания ($\sim A$) на алгебраическое выражение $1-A$. Пусть А – логическая переменная. Тогда, по определению операции отрицания, $\sim A=1$, если $A=0$, и $\sim A=0$, если $A=1$. Пусть А –обычная алгебраическая переменная. Тогда $1-A=1$, если $A=0$, и $1-A=0$, если $A=1$.

Таким образом, при подстановке одинаковых значений переменной А, выражения А и 1-А принимают одинаковые значения.

Рассмотрим замену импликации $(A \rightarrow B)$ на условие $B-A \geq 0$. По определению операции импликации, $A \rightarrow B$ – ложно, если $A=1$, а $B=0$; в остальных случаях $A \rightarrow B$ – истинно. Рассмотрим условие $B-A \geq 0$ при различных значениях А и В. Пусть $A=1$, $B=0$; тогда $B-A=-1$, и условие $B-A \geq 0$ ложно. Пусть $A=0$, $B=0$; тогда условие $B-A \geq 0$ истинно (так как $B-A=0$).
Аналогично можно показать, что условие $B-A \geq 0$ истинно, если $A=0$, $B=1$, или $A=1$, $B=1$.
Таким образом, логическая функция $A \rightarrow B$ и условие $B-A \geq 0$
истинны (или, наоборот, ложны) при одних и тех же значениях переменных А и
В.

В таблице 5.3 приведены еще некоторые правила представления условий, налагаемых на логические переменные.

Таблица 5.3

Условие	Алгебраическое представление
Хотя бы одна из логических переменных X1,X2,...,Xn принимает значение “истина”	$X1+X2+...+Xn \geq 1$
Только одна из логических переменных X1,X2,...,Xn принимает значение “истина”	$X1+X2+...+Xn=1$
Все логические переменные X1,X2,...,Xn одновременно принимают значение “истина”	$X1+X2+...+Xn=n$

Пример. Представим в алгебраической форме условия выбора лекарств

Рассмотрим условие $(A\& \sim B) \rightarrow C$. Воспользуемся правилами замены, приведенными в табл.5.2. Используя правило замены отрицания, получим:
$(A\&(1-B))\rightarrow C$. Выполнив замену конъюнкции, получим: $(A\cdot(1-B))\rightarrow C$. По правилу замены импликации получим: $C-(A\cdot(1-B)) \geq 0$.

Примечание. Промежуточные выражения $(A\&(1-B))\rightarrow C$ и $(A\cdot(1-B))\rightarrow C$ не являются вполне правильными с математической точки зрения, так как в них “смешаны” логические и алгебраические операции. Эти выражения приведены только для иллюстрации процесса перехода от логического к алгебраическому выражению.

Таким образом, условия выбора лекарств могут быть записаны в алгебраической форме следующим образом:
$C-(A\cdot(1-B)) \geq 0$
$A+B-A\cdot B-C \geq 0$
$1-C-A \geq 0$
$A+B+C \geq 0$
Чтобы выбрать подходящие лекарства (одно или несколько), требуется найти значения переменных A, B, C, удовлетворяющие этим условиям и принимающие значения 0 или 1.

Программа на языке Пролог для решения логической задачи

Приведем программу на Прологе, реализующую выбор лекарств, в соответствии с указанными выше условиями.

```
predicates
nondeterm poisk
nondeterm znach (integer)
goal
poisk
clauses
poisk:- znach(A), znach (B), znach (C),
C-(A*(1-B)) >= 0,
```

```
A+B-A*B-C >= 0,
1-C-A >= 0,
A+B+C >= 1,
write ("A=", A, " B=", B, " C=", C), !.
poisk:- write ("Заданы противоречивые условия").
znach(X):- X=0.
znach(X):- X=1.
```

При доказательстве целевого предиката poisk сначала происходит сопоставление предикатов znach (A), znach (B) и znach (C) с первым клозом предиката znach, т.е. znach (X):- X=0. В результате переменные A, B, C принимают значение 0. Затем происходит проверка условий. Для значений A=0, B=0, C=0 не выполняется условие $A+B+C \geq 1$. Поэтому происходит возврат к ближайшей "развилке"; в данном случае это предикат znach (C). Происходит сопоставление предиката znach (C) со вторым клозом предиката znach.
В результате переменная C принимают значение 1. Снова выполняется проверка условий. Процесс повторяется, пока не будут найдены значения переменных A, B, C, удовлетворяющие всем четырем условиям. Эти значения выводятся на экран. Предикат отсечения (!) указан только для того, чтобы второй клоз предиката poisk не сохранялся в памяти как нерассмотренная альтернатива. На выполнение данной программы предикат отсечения не влияет. Выполнив программу, получим: A=0, B=1, C=0. Таким образом, следует использовать лекарство B.

Примечание. Если бы условия задачи оказались противоречивыми, и никакие значения A, B, C не удовлетворяли бы всем условиям, то доказательство первого клоза предиката poisk завершилось бы неудачей. В этом случае был бы доказан второй клоз предиката poisk, выводящий на экран соответствующее сообщение.

Решение логических задач с использованием языка Пролог: заключительный пример

Пример. Требуется выполнить три вида работ (Р1, Р2, Р3). Имеется пять исполнителей (И1, И2, И3, И4, И5). При распределении исполнителей по работам необходимо учесть следующее:
• для работы Р1 требуются два исполнителя, для Р2 и Р3 – по одному (таким образом, один из исполнителей останется без работы); • каждый исполнитель может быть назначен только на одну работу;
• исполнители И2 и И5 не могут выполнять работу Р3;
• исполнителей И1 и И3 нельзя назначать на одну работу.

Для описания задачи введем переменные Xij, i=1,...,5, j=1,...,3. Эти переменные должны принимать значения Xij=1, если i-й исполнитель назначен на j-ю работу, и Xij=0 – в противном случае.
Первое, второе и третье условия удобно сразу записать в алгебраической форме.
Для работы Р1 требуются два исполнителя, для Р2 и Р3 – по одному:

$X11+X21+X31+X41+X51 = 2$
$X12+X22+X32+X42+X52 = 1$
$X13+X23+X33+X43+X53 = 1$

Каждый исполнитель может быть назначен только на одну работу:
$X11+X12+X13 \leq 1$

$X21+X22+X23 \le 1$
$X31+X32+X33 \le 1$
$X41+X42+X43 \le 1$
$X51+X52+X53 \le 1$

Исполнители И2 и И5 не могут выполнять работу Р3:
$X23 = 0$
$X53 = 0$

Условие невозможности назначения исполнителей И1 и И3 на одну работу запишем сначала с помощью логических функций:

$X11 \rightarrow X31$
$X12 \rightarrow X32$
$X13 \rightarrow X33$
По правилам, приведенным в табл.4.2, перейдем к алгебраической записи:

$(1-X31)-X11 \ge 0$
$(1-X32)-X12 \ge 0$
$(1-X33)-X13 \ge 0$

Требуется найти значения переменных Xij, i=1,...,5, j=1,...,3, соответствующие всем указанным условиям. Для этого используем следующую программу на Прологе.

```
predicates
nondeterm poisk
nondeterm znach (integer)
goal
poisk.
clauses
poisk:- znach(X11), znach (X12), znach (X13),
znach(X21), znach (X22), znach (X23),
znach(X31), znach (X32), znach (X33),
znach(X41), znach (X42), znach (X43),
znach(X51), znach (X52), znach (X53),
X11+X21+X31+X41+X51 = 2,
X12+X22+X32+X42+X52 = 1,
X13+X23+X33+X43+X53 = 1,
X11+X12+X13 <= 1,
X21+X22+X23 <= 1,
X31+X32+X33 <= 1,
X41+X42+X43 <= 1,
X51+X52+X53 <= 1,
X23 = 0, X53 = 0,
(1-X31)-X11 >= 0,
(1-X32)-X12 >= 0,
(1-X33)-X13 >= 0,
write("Вариант распределения: "), nl,
write("X11=", X11, " X12=", X12, " X13=", X13), nl,
write("X21=", X21, " X22=", X22, " X23=", X23), nl,
```

```
write("X31=", X31, " X32=", X32, " X33=", X33), nl,
write("X41=", X41, " X42=", X42, " X43=", X43), nl,
write("X51=", X51, " X52=", X52, " X53=", X53), nl,
readchar(_), fail.
poisk:- write("Все варианты найдены").
znach(X):- X=0.
znach(X):- X=1.
```

Работа этой программы аналогична предыдущей. Предикат fail, указанный в конце предиката poisk, требуется для того, чтобы найти все варианты решения.

Порядок выполнения работы

По заданию, выданному преподавателем, представить набор заданных утверждений с помощью логических функций. Перейти к алгебраическому представлению. Разработать и отладить программу на языке Пролог для решения полученной логической задачи.

Контрольные вопросы

1. Основные логические функции.
2. Представление логических функций в алгебраической форме.
3. Примеры представления знаний с помощью логических функций и в алгебраической форме.
4. Обработка знаний, представленных с помощью логических функций, в программах на Прологе.