ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение высшего образования
ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет автоматики и вычислительной техники
Кафедра «Электронные вычислительные машины»

Исчисление высказываний и исчисление предикатов

(Э2763)

Методические указания
к самостоятельным работам

Дисциплина «Логическое программирование»
Направление 230101.62 4 курс, дневное отделение

Киров, 2016

УДК 681.3.06.068

Методические указания для самостоятельных работ по курсу «Логическое программирование». / Вятский государственный университет. Киров, 2016, 20 с./ (Э2763).

Методические указания предназначены для студентов очного обучения специальности 230101.62 – Информатика и вычислительная техника.

Составитель к.т.н., доцент кафедры ЭВМ В.С. Ростовцев

1 Исчисление высказываний

Логическое программирование в широком смысле представляет собой семейство таких методов решения проблем, в которых используются приемы логического вывода для манипулирования знаниями, представленными в декларативной форме.

В узком смысле логическое программирование понимается как использование исчисления предикатов первого порядка в качестве основы для описания предметной области и осуществления логического вывода, называемого хорновским или резолюционным. Слово «логика» означает систематический метод рассуждений.

Исчисление высказываний (ИВ) изучает предложения, которые могут быть либо истинными, либо ложными.

Например. *Если верно, что когда идёт дождь, то дорога мокрая, то справедливо: если дорога сухая, то дождя нет.*

Здесь *«идёт дождь», «дорога мокрая»* являются *высказываниями*, т.е. предложениями, которые могут быть истинными и ложными.

Исчисление высказываний - это совокупность правил, используемых для определения истинности или ложности комбинации высказываний.

Высказывания или предложения- это просто утверждение, которые могут быть истинным или ложным.

Составное предложение или комбинация высказываний могут быть соединены различными связками (И, ИЛИ, НЕ, импликация, эквивалентность).

Таблица 1 - Перечень связок и их обозначение

Наименование связки	Обозначение связки	Тип связки	Другие обозначения
Отрицание	Not	Унарный	-,~, not,не
Конъюнкция	And	Бинарный	&,and,И
Дизъюнкция	Or	Бинарный	V, I, or,ИЛИ
Импликация	→	Бинарный	→, =>
Эквивалентность	<==>	Бинарный	⇔, <==>, ~

Таблица 2 - Таблица истинности выражений

P	Q	P V Q	P & Q	P→Q	P⇔Q
0	0	0	0	1	1
0	1	1	0	1	0
1	0	1	0	0	0
1	1	1	1	1	1

1.2 Теоремы булевой алгебры

1. Коммуникативный закон A&B=B&A
   AVB=BVA
2. Ассоциативный закон
   A&(B&C)=(A&B)&C
   AV(BVC)=(AVB) VC

3. Дистрибутивный закон
   A&(BVC)=(A&B)V(A&C)
   AV(B&C)=(AVB)&V(AVC)

4. Законы де Моргана
   ~(X & Y)= ~X V ~Y
   ~(X V Y)= ~X & ~Y

5. Свойства операций &,V
   A&T=A A& F=F
   A VT=T A V F=A
6. Свойства отрицания
   A & ~A=F
   A V ~A=T
7. Дополнение ~(~A)=A
8. Идемпотентность A &A=A A V A=A
9. Поглощение A & (AVB)=A
10. Полезные формулы преобразования

|= (A→B)= ~A V B

$\models$ $(A \Leftrightarrow B) = (A \rightarrow B) \& (B \rightarrow A)$

1.3 Правило модус поненс. Цепное правило

Широкое распространение получили два закона: модус поненс и модус толленс.

Закон модус поненс (Modus ponens)

$A, A \rightarrow B$

----------- (черта читается как следовательно)

B

Закон модус толленс (Modus tollens)

$\sim A, A \rightarrow B$

$\sim B$

Цепное правило

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

$(A \rightarrow C)$

$A, A \rightarrow B$

----------- (черта читается следовательно)

B

1.4 Исчисление высказываний и естественный язык

Идёт дождь (высказывание p), дорога мокрая (высказывание q).

Связка не коммуникативная.

«Если p, то q» можно представить

$p \rightarrow q$

Здесь p - это посылка, а q-это заключение.

Для удобства анализа рассуждений проводится переход к литерам (символам).

1.5 Выполняемые и общезначимые высказывания

Комбинацию высказываний (формулу) можно интерпретировать со значение ИСТИНА. В этом случае говорят, что формула выполняема.

Например, формула (p & q) выполняется, если p,q- истинны.

Формула (p & ~p) –невыполнимая.

Формула называется общезначимой или тавтологией, если она истинна независимо от входящих высказываний. Пример. ~(~p)=p

1.6 Методы дедукции

Дедукция (от лат. deductio выведение) – аналитический процесс, основанный на применении общих правил к частным случаям, с выводом результата, т.е. это умозаключение, в котором вывод про отдельный класс делается на основе абстрактного класса в целом, переход в процессе познания от общего к частному и единичному.

Прямая дедукция

H1&H2&…&Hn&~C=false

Обратная дедукция

~H1 V ~H2 V …~Hn V C=true
Hi-посылки; C- логическое следствие из Hi.

1.7 Алгоритм редукции

Применяется для доказательства общезначимости формул путём приведения к противоречию.

Рассмотрим пример.

[(p & q) → r] →[p →(q →r)]

Обозначим левую часть через литеру «A», а правую часть – через «B»
Допустим, что A→B = ложь
Тогда по таблице истинности (таблица 2)
A= [(p & q) → r] = истина (1)
B= [p →(q →r)] = ложь (0)
Из последнего следует, что p=1, (q →r)=0. Тогда q=1 r=0
Подставим полученные значения p=1, q=1 r=0
в выражение A
[(1&1)→0] и получим ложь, а должна быть 1.

Таким образом, пришли к противоречию. Следовательно, исходное выражение общезначимо.

Задача. Определить, общезначима ли формула [(p&q)→q] V (p→q) двумя методами: редукции и методом упрощения формулы с помощью законов булевой алгебры.

Метод редукции

1. Допустим вся формула ложна
   $[(p\&q)\to q]\ V\ (p\to q)=0$
2. Условие ложности выполняется, если правая и левая части выражения ложны.
3. Правая часть выражения ложна $(p\to q)=0$, только при $p=1$, $q=0$.
4. Подставим полученные значения $p=1$, $q=0$ в левую часть выражения.
   Левая часть принимает значение истины: $1\&0\to 0 = 1$.
5. Следовательно, получено противоречие. По предположению левая часть должна быть ложной, но она принимает истинное значение.
6. Вывод: данная формула не общезначима.

Метод упрощения формулы с помощью законов булевой алгебры

1. $[(p\&q)\to q]\ V\ (p\to q)=0$
2. Исключаем импликацию
   $[\sim(p\&q)vq]\ V\ (\sim pvq)=0$
3. Применяем закон де Моргана
   $(\sim p\ V\sim q\ V\ q)\ V\ (\sim p\ V\ q) = 0$
4. В полученном выражении $\sim q\ V\ q=1$
5. Применяя свойства булевой алгебры $A\ V\ 1=1$, получаем
   $[(p\&q)\to q]\ V\ (p\to q)\neq 0$. Вывод: формула не общезначима.

1.8 Метод резолюций

Дизъюнктом называется дизъюнкция конечного числа литер типа
$L1\ V\ L2\ V\ \ldots .V\ Ln$.

Пустой дизъюнкт – это единственный невыполнимый дизъюнкт, обозначаемый символом решетки (#).

Конъюнктивной нормальной формой (КНФ) называется конъюнкция конечного числа дизъюнктов.

Любая формула имеет логически эквивалентную КНФ.

Множество дизъюнктов невыполнимо тогда, когда пустой дизъюнкт типа $P\ V\ \sim P$ = # является логическим следствием из этого множества.

Не существует эффективного критерия для проверки выполнимости КНФ. Тем не менее есть удобный метод выявления невыполнимости множества дизъюнктов. Множество дизъюнктов невыполнимо тогда и только тогда, когда пустой дизъюнкт является логическим следствием из данного множества дизъюнктов.

Способ доказательства истинности утверждения A→B состоит в том, чтобы показать его отрицание ложно.

~(A→B)=~(~A V B)=A& ~B

Робинсон предложил применять для этого одно правило вывода, называемой *резолюцией.*

Если объединяются два высказывания (A) и (~A), то приходим к противоречию.

Метод резолюций

1. Приводим все посылки и отрицание заключения к конъюнктивной нормальной форме (КНФ). Для этого продвигаем отрицание внутрь, избавляемся от импликации, эквивалентности.
2. Посылки представляют собой конъюнкцию дизъюнктов. Выписываем каждый дизъюнкт с новой строки.
3. Каждый дизъюнкт- это дизъюнкция (может быть одночленная), состоящая из предложений и отрицаний предложений.
4. Выбирается два любые дизъюнкта, содержащие один и тот же атом, но с противоположными знаками.
5. Формируется резольвента, из которой удаляются атомы с противоположными знаками.
6. Процесс продолжается до получения пустого дизъюнкта, который выражает противоречие. Это завершает доказательство от противного (из P и ~P выводится ложь).

Пример применения метода резолюций

С использованием метода резолюций исчисления высказываний проверить выводимость цели (С) из логических предложений (П1-П4) с использованием трех стратегий логического вывода: предпочтение единичным элементам, опорного множества и «сначала вширь».

П1: Если я пойду завтра на первое занятие, то должен встать рано

ПЗ → ВР

П2: Если я пойду вечером на дискотеку, то лягу спать поздно.

ВД→ЛП

П3: Если я лягу спать поздно и встану рано, то буду вынужден довольствоваться 5 часами сна

ЛП & ВР →5ЧС

П4: Я просто не в состоянии довольствоваться 5 часами сна

~ 5ЧС

С: Я должен или не ходить на дискотеку или пропустить первое занятие

~ ПЗ V ~ ВД

Решение

1. Привести предложения к конъюнктивной нормальной форме
ПЗ → ВР = ~ ПЗ V ВР
ВД→ЛП = ~ ВД V ЛП
ЛП & ВР →5ЧС = ~ ЛП V ~ ВР V 5ЧС
~ 5ЧС
Отрицание цели ~(~ ПЗ V ~ ВД) = ПЗ & ВД
2. Выписать каждый ДИЗЪЮНКТ с новой строки
П1: ~ ПЗ V ВР
П2: ~ ВД V ЛП
П3: ~ ЛП V ~ ВР V 5ЧС
П4: ~ 5ЧС
П5: ПЗ
П6: ВД
3. Выполнить сопоставление дизъюнктов и формирование резольвент
7. R[П1, П5]= ВР
8. R[П3, П4] =~ ЛП V ~ ВР
9. R[П7, П8] =~ ЛП
10. R[П2, П6]= ЛП
11. R[П9, П10]= # (пустой дизъюнкт)

Получено противоречие, следовательно, цель С является логическим следствием предложений П1-П4.

1.9 Стратегии, используемые при доказательстве теорем с помощью метода резолюции

1. Стратегия опорного множества
2. Стратегия «сначала вширь»
3. Стратегия «предпочтение единичным элементам»

Стратегия опорного множества

Стратегия применима только при поиске доказательства

1. Некоторые предложения экспериментатор называют аксиомами, а все другие относятся к опорному множеству.
2. Программе запрещено проводить поиск между двумя аксиомами. Все другие резолюции допустимы.

Стратегия «сначала вширь»

1. Первоначально все предложения имеют уровень 0.
2. Стратегия порождает уровень 1 путём получения резольвент.
3. Из предыдущих уровней (уровни 0 и 1) стратегия порождает уровень 2 и т.д.

Стратегия «предпочтение единичным элементам»

1. Производится дедуктивный вывод предложений, содержащих возможно меньшее число литер.
2. Короткие предложения легче обрабатывать.
3. Стратегия даёт наивысший приоритет резолюциям единичных элементов.
4. Таким образом, стратегия устанавливает следующий порядок нахождения резольвент: единичный элемент с единичным элементом; единичный элемент с предложениями 2-го порядка и т.д.

1.10 Примеры задач

Задача 1. Установить выводимость заключения (C) из посылок (H1 – H3) тремя методами: прямой индукции, обратной индукции и методом резолюций.

$C=q\&\sim p$

$H1=p\to q$

$H2=(p\to\sim q)\&\sim r$

$H3=q\to r$

Задача 2. Определить, общезначима ли формула $[(\sim p\to q)\to r]\to(\sim r\to\sim q)$ двумя методами: метод редукций и метод упрощения формулы с помощью законов булевой алгебры.

2 Исчисление предикатов

Исчисление высказываний позволяет формализовать лишь малую часть множества рассуждений.

Например, рассмотрим рассуждение.

p: Все люди смертны

q: Сократ- человек *

r: Сократ- смертен

Это рассуждение правильное но оно выходит за рамки логики высказываний (в нем три высказывания).

Логика высказываний не позволяет корректно выразить рассмотренные рассуждения. Причина - логика высказываний моделирует высказывания и связки, позволяющие комбинировать высказывания. Первое высказывание содержит сцепную импликацию. " Быть человеком - значит быть смертным.

Текст можно переписать еще и так.

" Для всех X, если X является человеком, то X является смертным.

Сократ является человеком.

Следовательно, Сократ является смертным.

Здесь появились в дополнение к высказываниям и связкам новые компоненты: «для всех», «X», «Сократ», «является человеком» и « является смертным».

Исчисление предикатов не дает возможности выразить многие факты и рассуждения, которыми пользуются в жизни.

Например:

1. «Все ИМС, которые поставляет завод А стоят меньше 5000 руб.»
2. «Завод А поставляет ИМС К580ВВ55А»

Необходимо вывести, что ИМС К580ВВ55А стоит меньше 5000 руб.

Это нельзя сделать, так как текст второго предложения не входит непосредственно в первое предложение.

Исчисление высказываний можно применить, если представить предложение в следующем виде «Завод А поставляет ИМС К580ВВ55А, то ИМС К580ВВ55А стоят меньше 5000 руб.».

Умозаключение можно изобразить символически.

1. Завод А поставляет (К580ВВ55А) $\rightarrow$ стоит меньше (К580ВВ55А, 5000 руб.).
2. Завод А поставляет (К580ВВ55А).
3. По правилу модес поненс можно заключить, что «стоит меньше (К580ВВ55А, 5000 руб.)».

Выражение, стоящее перед скобками называется предикатом.
В общем случае предикат это функция от одного или нескольких аргументов с булевыми значениями истина и ложь.

2.1 Квантор общности ∀

Можно использовать переменные, входящие в предикаты, для сокращения конъюнкций, состоящего из большого числа схожих утверждений.
Завод поставляет (К580ВВ55А) → стоит меньше (К580ВВ55А, 5000)
Завод поставляет (К580ВИ53) → стоит меньше (К580ВИ53, 5000)
Завод поставляет (К580ВВ51) → стоит меньше (К580ВВ51, 5000)

Мы вводим переменную X и записываем всего одно утверждение
(∀ x) Завод поставляет (X) → стоит меньше (X , 5000).
Знак ∀ - квантор общности, который означает «для каждого» или «для всех».
При формальном прочтении: «для каждого X, если завод поставляет X, то X стоит меньше 5000».
Квантор управляет всей областью значений переменной, которая следует за ним. Если применяется квантор ∀ , то мы говорим что утверждение истинно для всех X в некотором множестве.
Или иными словами если подставить вместо X любое значение из некоторого множества, то утверждение будет истинно. Ограничение. Разрешается применять квантор ∀ только к переменным.
Недопустимо выражение: (∀ P,X) P(X) → ~ (~ P(X)).

2.2 Квантор существования ∃

Квантор ∃ применяется, когда нужно сказать что «существует хотя бы одно значение переменной». Читается «для некоторых X».
(∃ x) квадрат (X+1)=4 будет истинным при X=1 и X= -3.
Мы говорим что переменная связана квантором ∀ или ∃ в зависимости от того, какой из этих кванторов использован.
(∀ x) Завод поставляет (X) → (∃y) (стоит (x,y) & меньше (y,5000)).
Для любого X, если завод поставляет X, то найдется Y, такой, что X стоит Y рублей и Y меньше 5000.

2.3 Язык логики (исчисления) предикатов

Основные символы языка предикатов это переменные, индивидуальные константы, связки (~,&,V,→), квантор общности ∀ (для всех) и квантор существования ∃ (существует).
Синтаксис. Словарь позволяет определить термы, формы и квантификации, а также атомы и формулы.
Термом является всякая переменная и всякая функциональная форма. Функциональная форма - это функциональная константа, соединенная с подходящим числом термов.
Атом-это предикатная форма или некоторое равенство, выражение типа S=t , где S,t -термы.
Формула - атом есть формула, если A - формула то ~A -формула, A,B - формула, A & B, A V B, A →B, A⇔B - формула.

2.4 Представление знаний и рассуждений

Представление - это действие, делающее некоторое понятие воспринимаемым посредствам фигуры, записи, языка или формализма. Теория знаний изучает связи между субъектом и объектом.

Представление знаний - формализация истинных убеждений посредствам фигур, записей или языков. Нас особенно интересуют формализации воспринимаемые (распознаваемые ЭВМ). Они преобразуют наглядное представление (созданное посредствам речи, изображением, естественным языком, формальным языком типа алгебры и логики и т.д.)

Результат формализации - множество инструкций, составляющих часть языка программирования.

В искусственном интеллекте представление информации в ЭВМ должно иметь активный аспект: позволяющий не только запоминать, но и извлекать знания для рассуждений.

2.5 Синтаксис логики предикатов

Логика предикатов называется логикой первого порядка и допускает четыре типа выражений:

1. **Константы**. Они служат именами индивидуумов (в отличие от совокупностей). Константы представляются символами Женя-2, Пётр 1 и т.п.

Добавление 2 к слове Женя указывает на вполне определённого человека среди людей с такими именами.

2. **Переменные.** С их помощью обозначаются имена совокупностей таких как человек, книга, посылка, событие.
Символ Книга_22 (определенный экземпляр),
Книга - множество "всех книг"
Символами X,Y,Z представляются имена совокупностей (определенных множеств или понятий) Человек (X).

3. Предикатные имена. Они задают правила соединения констант и переменных, например, правило грамматики, процедуры, математические операции. Для предикатных имен используются символы типа: ПОСЫЛАТЬ, ПИСАТЬ, ПЛЮС, РАЗДЕЛИТЬ.

4. Функциональные имена. Они представляют такие же правила, как и предикаты. Чтобы не спутать с предикатными именами функциональные имена пишут одними строчными буквами: посылка, писать, плюс, разделить.
Символы, которые применяются для представления констант, переменных, предикатов и функций не являются словами русского языка.
Они суть символа представления - объективного языка «языка предикатов».

Комбинации кванторов

Если в утверждение входят несколько кванторов, то необходимо учитывать их взаимное расположение.
(∀x) (∀y) Меньше (x,y) → ((∀z) Меньше (y,z) → Меньше (x,z))

Здесь порядок первых двух кванторов несущественен, так как они одного типа.
В общем случае:
(∀x) (∀y) = (∀y) (∀x) = ∀ (x,y)

Эта формула записи короче и говорит о том, что порядок переменных x,y не важен. Мы можем также внести квантор по переменной z вперед, так как она не входит в предшествующий предикат Меньше(x,y).
(∀ x,y,z) Меньше (x,y) → (Меньше (y,z) → Меньше (x,y))

Переименование связанных переменных.

Однако мы не всегда можем выносить кванторы вперед, не меняя смысла утверждения. Рассмотрим утверждение.

$((\exists x)$ Завод поставляет (X)) & $((\exists x)$ Меньше (X,5000))

Это выражение нельзя просто переписать его в виде

$(\exists x)$ (Завод поставляет (X) & Меньше (X,5000)).

Здесь два вхождения переменной X в исходное утверждение относятся к разным объектам - они встречаются в независимых утверждениях. Тот «X», который поставляет завод не обязательно совпадет с тем «X», который меньше 5000. Мы можем преодолеть эту трудность, заменяя первую переменную на какую-то другую, например «y». Это допустимо, так как она похожа на локальную переменную в подпрограмме, ее область действия указывают внешние скобки , в которых содержится $((\exists x)$.

Внутри этих скобок можно заменять переменную на любую другую. Мы говорим, что X связанная переменная своим квантором.

$(((\exists y)$ Завод поставляет (У)) & $((\exists x)$ Меньше (X,5000)).

$((\exists\ x, y)$ Завод поставляет (У) & Меньше (X,5000).

В общем случае при вынесении кванторов следует переименовывать связанные переменные, если имеется опасность совмещения.

Выносить кванторы за скобки необходимо, но осторожно и внимательно.

Например, выражение $(\forall y)((\forall x) P(x)) \rightarrow R(x,y)$ не эквивалентно

$$(\forall y, x) P(x) \rightarrow R(x,y).$$

Это происходит потому, что в импликации как бы спрятано отрицание, которое действует на $(\forall x)$, но не $(\forall y)$.

2. 7 **Отрицание кванторов**

Рассмотрим пример

$$(\forall x) P(x) \rightarrow Q(x) \qquad P(x) \rightarrow Q(x) = \sim P(x) \vee Q(x)$$

Его отрицанием будет

$\sim((\forall x x \sim P(x) \vee Q(x))$ или $(\exists x) \sim (\sim P(x) \vee Q(x) = (\exists x) P(x) \ \&\ \sim Q(x)$

Заметим что кванторы $\forall$ и $\exists$ меняются друг на друга, связка ИЛИ меняется на связку И, а знаки литер меняются на противоположные.

Смысл этого преобразования. Если будем утверждать, что для всякого x истинность P(x) влечет истинность Q(x) («все красные яблоки-сладкие»), то контрпримером будет, когда имеется x для которого P(x) истинно, а Q(x) нет. «Есть яблоко красное, но не сладкое».

Основной приём при переводе с разговорного языка на язык исчисления предикатов - поиск слов вроде "всякий", "любой", "каждый", которые переходят в квантор всеобщности ∀ и слов "некоторый", "какой-то", которые переходят в квантор существования ∃.

2.8 Приведение к системе дизъюнктов

Любое выражение исчисления предикатов можно привести к конъюнктивной нормальной форме (КНФ). Этот процесс можно сравнить с упрощением алгебраического уравнения. Мы раскрываем скобки, умножает и можем привести подобные члены так, что получится сумма произведений переменных «x,y» и констант, упорядоченных по возрастанию степеней.

A + Bx + Cy + Dxx + Exy +.....

Аналогично можно используя закон дистрибутивности «раскрыть» формулу исчисления предикатов и получить коньюнкцию дизъюнктов, каждый из которых является литерой или дизъюнкцией литер.

Например:

A & (B & P(x)) & (C V Q(y)) & (E V P(x) V Q(y)) &...

Закон дистрибутивности.

1. A & (B V C)=(A & B) V (A & C)
2. A V (B & C)=(A V B) & (A V C)

Истинность конъюнкции эквивалентна истинности каждого дизъюнкта в отдельности, поэтому с такой формулой легче работать.

Это необходимо для применения метода резолюции.

Процесс приведения к нормальной форме:

1. Исключить эквивалентность P ⇔ Q = (P→Q) & (Q →P).
2. Исключить импликации P→Q = ~ P V Q.
3. Пронести внутрь знаки отрицания. Знаки отрицания проносятся внутрь скобок так, чтобы они оказались непосредственно перед атомами.

2.9 Принцип резолюций и доказательства теорем

Высказывания, которые исходно считаются истинными, называются *аксиомами или гипотезами,* а высказывания, которые следуют из них, называются *теоремами*.

Введение понятия согласуются с терминологией, используемой при описании подхода к математике, когда работа математика представляется как процесс получения новых теорем из хорошо аксиоматизированных областей, какими являются теория множеств и теория чисел.

Вопросы получения следствий для доказательства множества высказываний есть вопросы *доказательства теорем*.

В 60-х годах в этой области наблюдалась большая активность, связанная с возможностью использования ЭВМ для автоматического доказательства теорем. Именно эта область научной деятельности, по прежнему остается источником новых идей и методов которые легли в основу языка логического программирования Пролог.

Одним из фундаментальных достижений того времени явилось открытие Робинсоном принципа резолюций и его применение к автоматическому доказательству теорем.

Резолюция - это правило вывода, говорящее о том, как одно высказывание может быть получено из других.

Используя принцип резолюций можно полностью автоматически доказывать теоремы, выводя их из аксиом.

Необходимо лишь решить к каким из высказываний следует применять правило вывода, а правильные следствия из них будут строиться автоматически.

Правило резолюций разрабатывалось применительно к формулам, представленным в стандартной форме. Если заданы два дизъюнкта, связанные между собой определенным образом то это правило породит новый дизъюнкт, являющийся следствием двух первых.

Главная идея состоит в том, что если одна и та же формула появляется, как в левой части одного дизъюнкта, так и в правой части другого дизъюнкта, то дизъюнкт получаемый в результате соединения этих двух дизъюнктов, из которых вычеркнута повторяющаяся формула является следствием указанных дизъюнктов.

Ситуация усложняется, когда дизъюнкты содержат переменные. В этом случае две формулы не обязательно должны быть идентичными - они должны быть сопоставимы. Дизъюнкт, который получается из двух других с удалением повторяющихся формул, формируется с помощью некоторой операции. Эта операция включает в себя «конкретизацию» переменных до такой степени, чтобы две сопоставимые формулы были идентичными.

Упрощение состоит в том, что в общем случае, правило резолюций допускает сопоставление нескольких литералов в правой части одного дизъюнкта с несколькими литералами в левой части другого дизъюнкта.

Определение принципа резолюций

Определим теперь более точно принцип резолюции. Он сочетает в себе следующие две идеи:

Силлогизм - дедуктивное умозаключение, в котором из двух суждений следует новое суждение (заключение)

Принцип силлогизма.

1. Из A V B и Не A V C можно заключить B V C
2. Принцип отыскания частных случаев в исчислении предикатов.

Этот принцип устанавливает, что от формулы F (V1,V2,...Vn), которая по предположению справедлива для всех значений, входящих в нее переменных V1,V2,...Vn можно перейти к формуле F (t1,t2,...tk) соответственно путем подстановки термов t1,t2,...tk вместо переменных V1,V2,...Vn.

По определению термами может быть:

- индивидуальная константа, например «палец»;
- индивидуальная переменная, например X;
- функция других термов, например Q (X,У) и H (S,M(Q(S))).

Поясним теперь понятие общей фундаментальной резолюции. Дизъюнктивное предложение состоит из дизъюнкции литер. По определению литера является атомарной формулой или отрицанием ее. *Резольвента* (если она существует) литеры одного предложения и литеры другого предложения является следствием этих двух предложений.

Резольвента получается следующим образом:

1.Переменные переименовываются таким образом, чтобы все индивидуальные переменные в одном предложении отличались от всех индивидуальных переменных в другом.

2. Находиться минимальная подстановка, если таковая существует которая делает литеры одинаковыми, но противоположными по «знаку».
3. Выполняется подстановка повсеместно в обоих предложениях.
4. Если после подстановки одна и та же литера входит в предложение более чем один раз, в этом предложении вычеркиваются все одинаковые литеры, кроме одной.
5. Вычеркиваются две литеры, которые стали одинаковыми, но имеют противоположный знак.

Резольвента есть дизъюнкция литер, оставшихся в первом предложении, и литер, оставшихся во втором предложении.

Эффективность принципа резолюции означает, что при его использовании можно написать программу, которая за конечное число шагов найдет резольвенты любых двух предложений.

Принцип резолюции является полным как для нахождения доказательства, так и для отыскания следствий. Напомним, что цель процедуры нахождение доказательства состоит в том, чтобы показать что отрицание подлежащей доказательству теоремы неправомерно (приводит к противоречию).

Робинсон доказал, если конечное множество предложений несовместимо, то противоречие может быть обнаружено за конечное число применений принципа резолюции. Принцип резолюции является полным для нахождения следствий в смысле следующей теоремы.

В доказательства теорем стараются показать, что определенное правильно построенное выражение (ППВ) «В» есть логическое следствие множества ППВ S=A1,...Ак, называемых аксиомами рассматриваемой задачи. Правило вывода есть правило , при помощи которого из ранее полученных выражений можно получить новые.

2.10 Пример доказательства теорем с помощью метода резолюций

Теорема 1

P1 Если X есть часть V и если V есть часть Y, то X есть часть Y.

P2 Палец есть часть кисти руки.

P3 Кисть руки есть часть руки

P4 Рука есть часть человека

Цель: анализируя P1-P4, мы должны заключить, что палец есть часть человека.

Номер предложения	Предложение	Основание
P1	Часть (X,V)^Часть (V,Y)->Часть(X,Y)	Задано
P1	~[Часть (X,V)^Часть (V,Y)] V Часть(X,Y)= ~Часть (X,V) V ~Часть (V,Y)] V Часть(X,Y)	После исключения импликации. Символ ~ означает отрицание
P2	Часть (палец, кисть руки)	Задано
P3	Часть (кисть руки, рука)	Задано
P4	Часть (рука, человек)	Задано
P5(цель)	Часть (палец, человек)	Требуется доказать
P5	~Часть (палец, человек)	Отрицание цели
P6	~Часть (кисть руки,Y) **V** Часть(палец,Y)	R [P2,P1a] X= палец, V= кисть руки
P7	Часть (палец, рука)	R [P3,P6]
P8	~Часть (кисть руки, человек)	R[P5,P6]
P9	~Часть(X,палец) **V** Часть(X,кисть руки)	R [P2,P1b]
P10	~Часть(рука,Y) **V** Часть(Кисть руки,Y)	R[P3,P1a]
P11	Часть (кисть руки,человек)	R[P4,10a
P12	Пустой дизъюнкт (противоречие	R[P8,P11]

Стратегия опорного множества.

Экспериментатор называют некоторые (например заданные) предложения «аксиомами», а все другие предложения относят к «опорному множеству». Программа никогда не ищет резольвенту от двух аксиом.

Все другие резолюции допустимы. Она иногда более эффективна, чем стратегия и предпочтение единичным элементам.
В нашем примере P5 является единственным предложением, входящим в опорное множество. Стратегия " опорного множества " не будет пытаться подбирать для разрешения пары литер для аксиом P1-P4.

имя предл.	предложение	основание
P6	~Часть(палец,V)V~часть(V,человек)	r[P1c,P5]
P7	~Часть (кисть руки, человек)	r[P2,P6a]
P8	~Часть (палец, рука)	r[P4,P6в
P9	~Часть(кисть руки,V) V ~часть(V,чел)	r[P1c,P7]
P10	~Часть (рука, человек)	r[P3,P9a]
P11	Противоречие	

Исчисление предикатов мощнее, чем исчисление высказываний и дает возможность выразить многое из того, о чем мы хотели бы говорить или рассуждать. Используя предикаты кванторы, переменные и функциональные символы, мы можем выразить весьма сложные утверждения. Простые факты можно записать в виде атомов, состоящих из предикатов с константами в качестве аргументов. Таким образом, можно выражать различные знания, как общие аксиомы, так и факты, что важно для построения экспертных систем.

Литература

1. Грей П. " Логика, алгебры и базы данных /перевод с англ. Х.И. Килова. Под ред. Г.В. Орловского, А.О. Слисенко.-М.: Машиностроение, 1989.-368 с.

2. Логический подход к искусственному интеллекту: От классической логики к логическому программированию: Пер. с франц. /Тейз А. Грибомон, П. Луи Ж. и др. –М.: Мир, 1990.-432 с.