

СОДЕРЖАНИЕ

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №1. Основы логики высказываний

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №2. Метод резолюций в логике высказываний

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №3. Оценка сложности алгоритмов

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №4. Язык логики предикатов

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №5. Преобразование формул логики предикатов в клаузальную форму.

СПИСОК ЛИТЕРАТУРЫ

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1

Основы логики высказываний

Цель занятия: Получение практических навыков построения формул логики высказываний, анализа их свойств и преобразования в КНФ.

Теоретические сведения

Высказыванием в логике считается любое утверждение, относительно которого можно судить о его истинности или ложности [1]. Например, утверждение "Москва - столица России" является истинным высказыванием, а "Пингвины живут в Африке" - ложное высказывание. Вместе с тем, утверждение "В созвездии Кассиопеи существует жизнь", в соответствии с приведенным определением, высказыванием не является, т.к. мы не можем в настоящее время судить о его истинности или ложности.

Высказывание, не допускающее расчленения на более простые, называется элементарным. Для обозначения элементарных высказываний принято использовать символы p, q, r, s, t , которые в логике называют пропозициональными переменными. Составные высказывания строятся из элементарных с использованием пяти логических связок:

\neg - отрицание, соответствует отрицательной частице «не» в утверждениях естественного языка;

$\&$ - конъюнкция, соответствует союзу «и»;

\vee - дизъюнкция, соответствует союзу «или»;

\rightarrow - импликация, соответствует союзу «если ..., то ...»;

\leftrightarrow - эквивалентность, соответствует слову «эквивалентно», а также часто используемым в математике словосочетаниям «тогда и только тогда» или «необходимым и достаточным условием является».

С использованием пропозициональных переменных и логических связок высказывания представляются формулами. Например, утверждение «Если идет дождь и нет зонта, то поход в кино не состоится» записывается в виде формулы $(p \& \neg q) \rightarrow \neg r$. Здесь пропозициональные переменные соответствуют следующим элементарным утверждениям: p - «идет дождь»; q - «есть зонт»; r - «поход в кино состоится».

Формально синтаксис логических формул определяется следующими пра-

вилами:

1. Базис: всякая пропозициональная переменная является формулой.
2. Индукционный шаг: если X и Y - формулы, то $\neg X$, $(X \& Y)$, $(X \vee Y)$, $(X \rightarrow Y)$, $(X \leftrightarrow Y)$ - также формулы.
3. Ограничение: других формул нет.

А́ в данном определении символы X и Y являются метасимволами, они не принадлежат языку логики высказываний, а используются для обозначения произвольных формул. Если при построении формулы X формула Y используется как элемент в индуктивном правиле, то Y называется подформулой формулы X . Например, в формуле $(p \& q) \rightarrow (\neg r \vee t)$ можно последовательно выделить следующие подформулы: $(p \& q)$, $(\neg r \vee t)$, p , q , $\neg r$, t , r . Скобки используются для указания порядка выполнения операций и могут опускаться с учетом принятого старшинства операций: \neg , $\&$, \vee , \rightarrow , \leftrightarrow . Например, приведенная выше формула может быть записана в виде $p\&q \rightarrow \neg r\vee t$, а формула $p \leftrightarrow ((q \& r) \rightarrow \neg s)$ - в виде $p \leftrightarrow q\&r \rightarrow \neg s$.

Семантика формул определяется их свойством быть истинными или ложными, то есть принимать значение T (true) - истина, или F (false) - ложь. Интерпретировать формулу логики высказываний, значит приписать значения истинности составляющим ее элементарным высказываниям и вычислить истинностное значение всей формулы. При этом логические связки рассматриваются как функции на множестве истинностных значений (булевы функции), определенные в соответствии с табл. 1.

Интерпретация, при которой формула принимает значение T , называется моделью этой формулы.

Формула выполнима (непротиворечива), если она допускает хотя бы одну модель, т.е. ее можно интерпретировать со значением T . Например, формула $p \& q$ является выполнимой.

Таблица 1

X	Y	$\neg X$	$X \& Y$	$X \vee Y$	$X \rightarrow Y$	$X \leftrightarrow Y$
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

Формула невыполнима (противоречива, тождественно ложна), если она не имеет ни одной модели, то есть при всех интерпретациях является ложной. Например, формула $(p \& \neg p)$ невыполнима.

Формула общезначима (является тавтологией), если она истинна в любой интерпретации. Например, формула $(p \vee \neg p)$ общезначима.

Формулы равносильны (логически эквивалентны), если они имеют одинаковые таблицы истинности.

Пример. Записать следующее утверждение в виде формулы логики высказываний и определить выполнимость, общезначимость и количество моделей этой формулы:

«Студент не допускается к экзамену тогда и только тогда, когда у него не сдана курсовая работа или есть задолженности по лабораторным работам».

Решение. Обозначим все встретившиеся элементарные высказывания пропозициональными переменными:

p - «студент допущен к экзамену»;

q - «курсовая работа сдана»;

r - «есть задолженность по лабораторным работам».

Тогда формула запишется в следующем виде: $\neg p \leftrightarrow (\neg q \vee r)$. Построим таблицу истинности, вычисляя сначала значения подформул (Табл. 2).

Таблица 2

p	q	r	$\neg p$	$\neg q$	$\neg q \vee r$	$\neg p \leftrightarrow (\neg q \vee r)$
F	F	F	T	T	T	T
F	F	T	T	T	T	T
F	T	F	T	F	F	F
F	T	T	T	F	T	T
T	F	F	F	T	T	F
T	F	T	F	T	T	F
T	T	F	F	F	F	T
T	T	T	F	F	T	F

Таким образом, данная формула выполнима, т.к. имеет модели и не общезначима, т.к. имеются интерпретации, при которых она ложна. Формула имеет 4 модели.

Литерой (литералом) в логике высказываний называется любая пропозициональная переменная с отрицанием или без него. Дизъюнкт - дизъюнкция конечного числа литер. Следующие формулы являются примерами дизъюнктов: $(p \vee \neg q \vee r)$, $\neg t$, $(\neg s \vee q)$. Любой дизъюнкт, содержащий хотя бы одну литеру, является выполнимой формулой. Единственным невыполнимым дизъюнктом является пустой дизъюнкт, то есть дизъюнкт не содержащий ни одной литеры. Поэтому его принято обозначать логической константой F.

Конъюнктивной нормальной формой (КНФ) называется конъюнкция конечного числа дизъюнктов. Приведенной нормальной формой называется нормальная форма, в которой опущены дизъюнкты, содержащие контрарные пары (тавтологии) и повторения литер в пределах одного и того же дизъюнкта.

Любая формула может быть преобразована в логически эквивалентную ей КНФ с использованием следующего алгоритма:

1. Исключить из формулы все связки импликации и эквивалентности, используя следующие тождества:

$$A \rightarrow B \equiv \neg A \vee B ;$$

$$A \leftrightarrow B \equiv (A \rightarrow B) \& (B \rightarrow A) \equiv (\neg A \vee B) \& (\neg B \vee A).$$

2. Сузить области действия отрицаний и исключить двойные отрицания.

Для этого используются законы де Моргана и снятия двойных отрицаний:

$$\neg(A \& B) \equiv \neg A \vee \neg B ;$$

$$\neg(A \vee B) \equiv \neg A \& \neg B ;$$

$$\neg\neg A \equiv A.$$

3. Применить необходимое число раз правило дистрибутивности дизъюнкции относительно конъюнкции:

$$(A \& B) \vee C \equiv (A \vee C) \& (B \vee C).$$

Пример. Преобразовать в КНФ следующую формулу логики высказываний: $(p \& \neg q) \leftrightarrow (q \vee \neg s)$.

Решение. Выполним преобразования в соответствии с приведенным выше алгоритмом.

1. Исключение связки эквивалентности:

$$[(p \& \neg q) \rightarrow (q \vee \neg s)] \& [(q \vee \neg s) \rightarrow (p \& \neg q)] =$$

$$[\neg(p \& \neg q) \vee (q \vee \neg s)] \& [\neg(q \vee \neg s) \vee (p \& \neg q)].$$

2. Сужение области действия отрицаний и снятие двойных отрицаний:

$$[(\neg p \vee \neg\neg q) \vee (q \vee \neg s)] \& [(\neg q \& \neg\neg s) \vee (p \& \neg q)] =$$

$$[(\neg p \vee q \vee q \vee \neg s)] \& [(\neg q \& s) \vee (p \& \neg q)] =$$

$$[(\neg p \vee q \vee \neg s)] \& [(\neg q \& s) \vee (p \& \neg q)].$$

3. Применение дистрибутивности, исключение тавтологий:

$$[(\neg p \vee q \vee \neg s)] \& [(\neg q \vee p \& q) \& (s \vee p \& \neg q)] =$$

$$[(\neg p \vee q \vee \neg s)] \& [(\neg q \vee p) \& (\neg q \vee q) \& (s \vee p) \& (s \vee \neg q)] =$$

$$(\neg p \vee q \vee \neg s) \& (\neg q \vee p) \& (s \vee p) \& (s \vee \neg q).$$

Получили КНФ.

Упражнения

1. Для приведенных формул логики высказываний построить соответствующие им логические функции в виде таблиц истинности, определить общезначимость, выполнимость (невыполнимость) и число моделей формулы:

а) $\neg s \rightarrow (\neg t \rightarrow r)$;

б) $\neg q \leftrightarrow (p \rightarrow \neg r \vee s)$;

в) $\neg p \rightarrow \neg((q \& p) \rightarrow (p \vee r))$;

г) $p \& (q \vee \neg p) \& ((\neg q \rightarrow p) \rightarrow q)$;

д) $\neg(r \& \neg q \vee p) \rightarrow \neg t$;

е) $\neg(p \& q) \leftrightarrow \neg q$;

ж) $\neg(r \& \neg t) \vee \neg(p \vee q)$;

з) $(q \vee \neg(p \& t)) \rightarrow ((q \& \neg t) \rightarrow p)$.

2. Записать следующие утверждения в виде формул логики высказываний, построить таблицы истинности и определить общезначимость, выполнимость (невыполнимость) и число моделей полученных формул:

а) Если Сидоров поедет на автобусе, то его уволят с работы, если автобус опоздает.

б) Необходимое и достаточное условие для жизни растений состоит в наличии питательной почвы, чистого воздуха и солнечного света.

в) Если «Торпедо» или «Динамо» проиграют, а «Локомотив» выиграет, то «Спартак» потеряет первое место.

г) Если рабочие или администрация упорствуют, то забастовка будет урегу-

лирована тогда и только тогда, когда правительство добьется судебного запрещения, но войска не будут посланы на завод.

д) Если вечером будет туман или снег, то Джон или останется дома или должен будет взять такси.

е) Либо рост инфляции эквивалентен снижению уровня жизни, либо рост производства влечет то, что уровень жизни не снижается.

ж) Если будет идти дождь или снег, то футбольный матч либо не состоится, либо его результат не будет отражать соотношение сил.

з) Если животное млекопитающее и имеет острые зубы и имеет клыки и не ест траву, то это хищник.

3. Преобразовать следующие формулы в КНФ:

а) $(\neg r \ \& \ \neg t) \leftrightarrow (p \vee q)$;

б) $(r \ \& \ \neg q \ \& \ \neg t) \rightarrow (t \vee q \vee r)$;

в) $((p \rightarrow q) \rightarrow (r \rightarrow \neg p)) \rightarrow (\neg q \rightarrow \neg r)$;

г) $(q \rightarrow (p \rightarrow t)) \rightarrow ((q \rightarrow \neg t) \rightarrow (q \rightarrow \neg p))$;

д) $\neg(r \vee q \vee \neg p) \rightarrow \neg(t \rightarrow \neg s)$;

е) $\neg(p \ \& \ q) \leftrightarrow \neg(r \vee q)$;

ж) $\neg(r \ \& \ \neg t) \vee \neg(p \vee q)$.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2

Метод резолюций в логике высказываний

Цель занятия: Практическое освоение метода резолюций в логике высказываний для решения проблемы дедукции.

Теоретические сведения

Пусть E - множество формул. Формулу C называют **логическим следствием** из E , если при всех интерпретациях, при которых одновременно истинны все формулы из E , формула C также истинна. Этот факт записывается следующим образом $E \models C$.

Основной проблемой логики является **проблема дедукции** [1, 5], которая заключается в том, чтобы для заданной формулы C и множества формул E определить, является ли формула C логическим следствием из E . Непосредственное решение проблемы дедукции методом перебора всевозможных интерпретаций характеризуется высокой вычислительной сложностью, так как при n пропозициональных переменных число интерпретаций равно 2^n . Проблема дедукции может быть сведена к задаче установления невыполнимости множества формул с помощью **принципа дедукции**: формула C является логическим следствием конечного множества E формул тогда и только тогда, когда множество $E \cup \{\neg C\}$ невыполнимо. Множество формул невыполнимо, если не существует интерпретации, при которой все его элементы одновременно истинны, то есть формулы множества не имеют общей модели. Таким образом, с точки зрения свойства вы-

полноты, множество формул эквивалентно конъюнкции своих элементов.

Вместе с тем, множество формул невыполнимо тогда и только тогда, когда его логическим следствием является константа F. Поэтому принцип дедукции можно записать в символической форме следующим образом:

$$\{H_1, \dots, H_n\} \models C \leftrightarrow \{H_1, \dots, H_n, \neg C\} \models F.$$

Формулы H_i называют гипотезами или посылками, формулу C - заключением.

Невыполнимость множества дизъюнктов может быть эффективно установлена с помощью метода резолюций.

Литерал и его отрицание называют контрарной парой. Например, p и $\neg p$ - контрарная пара.

Правило резолюций: если два дизъюнкта содержат контрарную пару, то их логическим следствием является дизъюнкт, полученный объединением исходных дизъюнктов, из которых исключены литералы контрарной пары. Этот новый дизъюнкт называют **резольвентой** исходных (родительских) дизъюнктов.

Например, резольвентой дизъюнктов $(p \vee q \vee r)$ и $(\neg p \vee s)$ является дизъюнкт $(q \vee r \vee s)$.

Резольвентой двух однолитеральных дизъюнктов может быть только пустой дизъюнкт F. Известно [1, 2], что для любого невыполнимого множества дизъюнктов за конечное число применений правила резолюций может быть получен пустой дизъюнкт. Так как, в соответствии с принципом дедукции, проблема логического следования сводится к задаче установления невыполнимости множества формул, метод резолюций позволяет решить основную проблему логики - проблему дедукции.

Пусть S - исходное множество дизъюнктов, s_1 и s_2 - дизъюнкты множества S, l - литерал. Тогда алгоритм доказательства невыполнимости множества S дизъюнктов на основе метода резолюций в общем виде записывается следующим образом:

While $F \notin S$ do

 выбрать l, s_1, s_2 такие, что $l \in s_1$ и $\neg l \in s_2$,

 вычислить резольвенту r дизъюнктов s_1 и s_2 ;

 заменить S на $S \cup \{r\}$.

end.

Данный алгоритм предполагает перебор дизъюнктов. Для организации перебора в методе резолюций используются различные стратегии. Рассмотрим три стратегии: насыщения уровня, линейную и предпочтения одночленам.

Пусть S - исходное множество дизъюнктов. Обозначим $R(S) = R_1(S)$ - объединение S с множеством всех резольвент порожденных от дизъюнктов S. Тогда $R_n(S) = R(R_{n-1}(S))$, значение n назовем уровнем (или глубиной) опровержения. Стратегия насыщения уровня предполагает последовательное порождение всех резольвент уровня 1, затем уровня 2 и т.д. до получения пустого дизъюнкта. Таким образом, данная стратегия является по существу стратегией поиска в ширину.

Если резольвента, полученная на i-ом шаге вывода ($i > 0$), всегда имеет в качестве одного из своих родительских дизъюнктов (называемого левым) дизъ-

юнкты, полученный на $(i-1)$ -ом шаге вывода, то такой вывод называется линейным.

В стратегии предпочтения одночленам перебор дизъюнктов-претендентов на резолюцию выполняется в порядке возрастания их длины. Вначале делается попытка построить резольвенты между одночленами, т.е. дизъюнктами, содержащими единственный литерал. Если это удастся, то сразу получается опровержение, если нет - то ищутся резольвенты для пар одночлен-двучлен, затем двучлен-двучлен и т.д. Перебор пар дизъюнктов в данной стратегии должен выполняться таким образом, чтобы сумма длин родительских дизъюнктов в процессе перебора не убывала.

Полное решение проблемы дедукции с использованием метода резолюций включает следующие шаги:

1. Запись исходного рассуждения (посылок и заключения) в виде логических формул.
2. Добавление отрицания заключения к множеству посылок.
3. Преобразование всех формул в КНФ.
4. Доказательство невыполнимости полученного множества дизъюнктов методом резолюций.

Задача. Записать формально на языке логики высказываний следующее рассуждение и доказать его справедливость, используя метод резолюций:

Посылки: Если налоги в бюджет не собраны, то либо секвестрируется бюджет, либо правительство уходит в отставку. Если секвестрируется бюджет, то падает уровень жизни. Налоги в бюджет не собраны.

Заключение: Либо падает уровень жизни, либо уровень жизни не падает и правительство уходит в отставку.

Решение.

1. Введем символические обозначения элементарных высказываний:

h - «налоги в бюджет не собраны»;

p - «бюджет секвестрируется»;

q - «правительство уходит в отставку»;

r - «уровень жизни падает».

Запишем данное рассуждение формально в виде логического следования:

$$\{ h \rightarrow (p \vee q), p \rightarrow r, h \} \models (r \vee \neg r \& q)$$

2. Добавим отрицание целевого утверждения к множеству посылок:

$$\{ h \rightarrow (p \vee q), p \rightarrow r, h, \neg(r \vee \neg r \& q) \}$$

3. Преобразуем все формулы в КНФ:

$$h \rightarrow (p \vee q) = \neg h \vee p \vee q;$$

$$p \rightarrow r = \neg p \vee r;$$

$$\neg(r \vee \neg r \& q) = \neg r \& \neg(\neg r \& q) = \neg r \& (r \vee \neg q).$$

Итак, имеем следующее множество посылок:

$$\{ (\neg h \vee p \vee q), (\neg p \vee r), h, \neg r, (r \vee \neg q) \}.$$

4. Продолжим перебор с использованием стратегии предпочтения одночленам. В начале выпишем и пронумеруем дизъюнкты исходного множества:

- 1) $\neg h \vee p \vee q$

- 2) $\neg p \vee r$
- 3) h
- 4) $\neg r$
- 5) $r \vee \neg q$
- 6) $\neg p$ (4, 2)
- 7) $\neg q$ (4, 5)
- 8) $p \vee q$ (3, 1)
- 9) q (6, 8)
- 10) F (7, 9)

Справедливость рассуждения доказана. Рассмотрим теперь вывод с использованием линейной стратегии:

- 6) $\neg h \vee q \vee r$ (1, 2)
- 7) $q \vee r$ (6, 3)
- 8) q (7, 4)
- 9) r (8, 5)
- 10) F (9, 4)

Упражнения

1. Используя метод резолюций доказать невыполнимость (или выполнимость) следующих множеств дизъюнктов. Применять произвольный порядок перебора дизъюнктов, а также, по указанию преподавателя, одну из следующих стратегий: предпочтение одночленам, линейную, насыщение уровня.

- а) $\{(p \vee q \vee \neg s), (p \vee \neg q), s, \neg p\}$;
- б) $\{(q \vee r), (q \vee \neg r), (r \vee \neg q), \neg r\}$;
- в) $\{(p \vee q \vee \neg s), (q \vee \neg p), (\neg q \vee \neg s), s\}$;
- г) $\{(q \vee \neg r), (\neg q \vee \neg r), (q \vee r), (\neg p \vee \neg r), \neg q\}$;
- д) $\{(p \vee r), (p \vee \neg r), (\neg p \vee \neg r), (q \vee \neg p \vee \neg r), \neg p\}$;
- е) $\{(r \vee \neg q \vee s), (\neg q \vee \neg r \vee \neg s), (q \vee \neg s), s, \neg r\}$;
- ж) $\{(q \vee s \vee \neg p), (s \vee \neg q), (\neg p \vee \neg s), (r \vee \neg p), p\}$;
- з) $\{(p \vee q \vee r), (p \vee r \vee \neg q), (r \vee \neg p \vee \neg q), (\neg q \vee \neg r), q\}$;
- и) $\{(p \vee s), (p \vee r \vee \neg s), (r \vee \neg p \vee \neg s), (\neg r \vee \neg s), s\}$;
- к) $\{(q \vee \neg p \vee \neg r), (\neg p \vee \neg q), (p \vee q), (q \vee r), \neg q\}$.

2. Записать формально следующее рассуждение на языке логики высказываний и доказать его справедливость, используя метод резолюций.

а) Посылки: Если идет дождь, то нежарко. Если светит солнце, то жарко. Идет дождь.

Заключение : Нежарко и не светит солнце.

б) Посылки: Экзамен сдан вовремя или сессия продлена. Если сессия продлена, то не сдана курсовая работа или не зачтены лабораторные работы. Курсовая работа сдана. Экзамен вовремя не сдан.

Заключение: Неверно, что если курсовая работа сдана, то лабораторные работы зачтены.

в) Посылки: Если имеет место денежная эмиссия, то растет курс доллара. Если эмиссии нет и инфляция не растет, то курс доллара не растет. Инфляция не растет.

Заключение: Имеет место эмиссия и растет курс доллара или нет эмиссии и курс доллара не растет.

г) Посылки: Заработная плата возрастет только, если будет инфляция. Если будет инфляция, то увеличится стоимость жизни. Заработная плата возрастет.

Заключение: Стоимость жизни увеличится.

д) Посылки: Если 2 - простое число, то это наименьшее простое число. Если 2 - наименьшее простое число, то 1 не есть простое число. Число 1 не есть простое число.

Заключение: 2 - простое число.

е) Посылки: Джон или переутомился или он болен. Если он переутомился, то он раздражается. Он не раздражается.

Заключение: Джон болен.

ж) Посылки: Если завтра будет холодно, я надену теплое пальто, если рукав будет починен. Завтра будет холодно, а рукав не будет починен.

Заключение: Я не надену теплое пальто.

з) Посылки: Если исход скачек будет предрешен сговором или в игорных домах будут орудовать шулеры, то доходы от туризма упадут и город пострадает. Если доходы от туризма упадут, полиция будет довольна. Полиция никогда не бывает довольна.

Заключение: Исход скачек не предрешен сговором.

и) Или Сэлли и Боб одного возраста, или Сэлли старше Боба. Если Сэлли и Боб одного возраста, то Нэнси и Боб не одного возраста. Если Сэлли старше Боба, то Боб старше Уолтера.

Заключение: Или Нэнси и Боб не одного возраста, или Боб старше Уолтера.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3

Оценка сложности алгоритмов

Цель занятия: Сформировать и закрепить практические навыки оценки временной и емкостной сложности алгоритмов.

Теоретические сведения

Основными мерами сложности алгоритма являются время, необходимое для его реализации, и объем требуемой памяти [6, 7]. При этом каждому классу задач можно сопоставить число, характеризующее объем входных данных, которое принято называть размером входа. Например, в задаче логического вывода на основе метода резолюций размером входа следует считать число дизъюнктов исходного множества, в процедуре поиска контрарной пары в двух дизъюнктах - число литералов в этих дизъюнктах и т.д. Очевидно, что сложность алгоритма является функцией размера входа.

Временной сложностью алгоритма называется время, затрачиваемое на

вычисление результата, выраженное как функция размера входа. Емкостная сложность - общая емкость памяти, использованной в процессе реализации алгоритма, как функция размера входа.

Сложность алгоритма может быть различной и при фиксированном размере входа, но различных входных данных. Например, при поиске контрарной пары в двух дизъюнктах фиксированной длины, контрарные литералы могут оказаться первыми или последними в этих дизъюнктах. Очевидно, что время поиска контрарной пары в этих случаях окажется различным. Поэтому различают сложность в худшем и в среднем случае. Сложность в худшем случае - это максимальная при данном размере входа сложность. Средняя сложность - средняя по всем входам данного размера сложность.

Временная сложность в худшем случае - это функция $T(n)$, равная максимальной (по всем входам размера n) из сумм времен, затраченных на каждую сработавшую команду (оператор).

Емкостная сложность в худшем случае - это функция $S(n)$, равная максимальной (по всем входам размера n) из сумм емкостей всех ячеек (элементов) памяти к которым было обращение.

Для точного теоретического определения временной и емкостной сложности требуется знать времена выполнения каждой команды (оператора) и объем памяти, используемый каждой переменной. Так как эти значения зависят от конкретной вычислительной машины, в теории сложности алгоритмов принято оценивать порядок сложности.

Говорят, что функция $f(n)$ имеет сложность порядка $g(n)$, если существуют такие положительные константы c и n_0 , что для всех $n \geq n_0$, $f(n) \leq c * g(n)$ [7]. Этот факт записывается так: $f(n)$ есть $O(g(n))$. Обычно рассматриваются положительные функции натуральных чисел. Например, функция $f(n) = 10 * n^3$ есть $O(n^3)$, т.к. существуют $c=10$ и $n_0 = 1$, для которых $10 * n^3 \leq 10 * n^3$. Неверно, что $10 * n^3$ есть $O(n^2)$, так как для любого сколь угодно большого наперед заданного значения константы c найдется такое значение n_0 аргумента, что $f(n) \geq c * g(n)$ для всех $n \geq n_0$.

При оценке порядка временной сложности определяется зависимость числа шагов в алгоритме от размера входа. Шагом считается таёёё фрагмент алгоритма, время реализации которого не зависит от размера входа, а число реализаций этого фрагмента зависит от размера входа. Например, шагом является тело цикла, независимо от числа содержащихся в нем операторов, если оно не содержит вложенных циклов, а параметр цикла связан с размером входа.

При определении порядка емкостной сложности, в качестве единицы необходимо выбрать некоторый элементарный в данном классе алгоритмов объект. Таким объектом в задаче логического вывода на основе метода резолюций следует считать литерал. При этом точный объем памяти (в байтах), занимаемой одним литералом, зависит от выбранных типов и структур данных и не влияет на порядок сложности.

При теоретическом анализе порядка сложности алгоритмов принято оцени-

вать сложность в худшем случае. Теоретическая оценка средней сложности, как правило, весьма затруднительна, т.к. предполагает учет всевозможных комбинаций входных данных и вероятностей их появления. Она может быть получена (в некотором приближении) экспериментальным путем с помощью программно реализованного алгоритма [***].

При оценке сложностей можно использоваться различные критерии.

Равномерный весовой критерий предполагает, что каждая команда (оператор) затрачивает одну единицу времени и каждая переменная в алгоритме занимает одну ячейку памяти, независимо от принимаемых ей значений.

Логарифмический весовой критерий учитывает ограниченность размера реальной ячейки памяти. Пусть $l(i)$ - логарифмическая функция на целых числах, определенная следующим образом:

$$l(i) = \begin{cases} \lceil \log_2 |i| \rceil + 1, & \text{при } i \neq 0; \\ 1, & \text{при } i=0. \end{cases}$$

Здесь [...] - округление до меньшего целого.

Логарифмический критерий при оценке временной сложности основан на допущении, что время выполнения команды зависит от величины обрабатываемых операндов и определяется значением $l(i)$, где i - величина обрабатываемого данной командой операнда.

Логарифмический критерий при оценке емкостной сложности учитывает размер ячейки (в битах) в зависимости от величины операнда. Логарифмическая емкостная сложность - сумма по всем использованным ячейкам величин $l(x_i)$, где x_i - наибольшее по абсолютной величине значение, содержащееся в i -ой ячейке за все время вычислений.

Алгоритм может иметь существенно различные временные и емкостные сложности в зависимости от используемого весового критерия.

Пример. Определить временную и емкостную сложность алгоритма сортировки массива по возрастанию методом «пузырька» при равномерном и логарифмическом весовых критериях.

Решение. В соответствии с данным методом для всех элементов массива, начиная со второго, последовательно выполняется «всплытие», при котором текущий элемент, если он меньше предыдущего, меняется с ним местами. Обозначим i - текущий номер элемента при просмотре массива в глубину, j - номер текущего элемента при "всплывании" (процедура `elem_up`), $A(j)$ - j -ый элемент массива, n - число элементов в массиве. Используя синтаксис языка Паскаль, можно записать:

```

program Sort;
procedure elem_up (i: integer); {процедура «всплытие» элемента}
begin
  j := i
  while (j > 1) and (A(j) < A(j-1)) do
  begin
    X:= A(j);           {перестановка соседних элементов}

```

```

    A(j):=A(j-1);
    A(j-1):=X;
    j := j-1;
end
end;
begin          {основной цикл программы}
  for i := 2 to n do elem_up(i);
end.

```

Временная сложность при равномерном весовом критерии. Очевидно, что размером входа для данного алгоритма следует считать число n элементов в массиве, а шагом алгоритма - тело цикла в процедуре «всплывание». В теле основной программы имеем $(n-1)$ проходов по циклу. В процедуре `elem_up` ("всплывание" элемента) - в худшем случае выполняется i проходов по циклу. Тогда, общее число шагов в алгоритме рѳаѳѳ:

$$\sum_{i=2}^n i = (n+2)*(n-1)/2 = (n^2+n-2)/2.$$

Таким образом, временная сложность алгоритма при равномерном весовом критерии есть $O(n^2)$.

Временная сложность при логарифмическом весовом критерии. Логарифмическая временная сложность операций инкрементации и сравнения с граничным значением параметра цикла в основной программе на i -ом проходе есть $\log(i)$. Суммарная сложность этих операций на всех проходах основной программы:

$$\sum_{i=2}^n \log(i) = \log(n!)$$

В процедуре `elem_up` логарифмическая временная сложность складывается из двух частей. Во-первых, работа со счетчиком "всплывания" j . Так как он выполняет обратный счет от i до 1, то временная сложность этой операции для i -го прохода цикла основной программы есть:

$$\log(i) + \log(i-1) + \dots + \log 1 = \log(i!).$$

За время всех n проходов:

$$\sum_{i=2}^n \log(i!) = \log(\prod_{i=2}^n i!).$$

Во-вторых, работа с элементами массива (их сравнение и перестановка). Логарифмическая сложность этой операции определяется значениями элементов массива и равна $\log(A_{\max})$, где A_{\max} - максимальное значение элементов массива. Число шагов на i -ом проходе основной программы в худшем случае равно $i*\log(A_{\max})$. За все проходы имеем:

$$\sum_{i=2}^n (i*\log(A_{\max})) = (n*(n+1)/2)*\log(A_{\max}).$$

Тогда суммарная временная сложность при логарифмическом критерии есть:

$$\log(n!) + \log\left(\prod_{i=1}^n (i!)\right) + (n*(n+1)/2)*\log(A_{\max}).$$

Окончательно для временной сложности при логарифмическом весовом критерии имеем

$$O(\text{MAX}(\log(\prod_{i=1}^n (i!)), n^2*\log(A_{\max}))).$$

Емкостная сложность при равномерном весовом критерии очевидно определяется размером n массива, поэтому есть $O(n)$.

Емкостная сложность при логарифмическом весовом критерии. Определяется с одной стороны размером массива и его элементов, а с другой - емкостью счетчика:

$$\log(n) + n*\log |A_{\max}|.$$

Окончательно имеем $O(n*\log |A_{\max}|)$.

Упражнения

1. Определить временную и емкостную сложность алгоритмов решения следующих задач при равномерном и логарифмическом весовых критериях.

- а) Вычислить $n!$.
- б) В массиве ϵ n целых чисел найти все пары элементов, сумма которых четна и сформировать новый массив из этих сумм.
- в) Вычислить n^n .
- г) Вычислить среднее значение элементов массива из n чисел.
- д) Вычислить число размещений $A(n, m) = n! / (m! * (n - m)!)$.
- е) Вычислить максимальное значение в массиве чисел.
- ж) \hat{A} множестве из n дизъюнктов найти и исключить все надслучаи.
- з) \hat{A} множестве из n дизъюнктов найти и исключить все тавтологии.
- и) В множестве из n дизъюнктов найти и исключить все дизъюнкты с уникальными литералами.
- к) В массиве ϵ n целых чисел найти все тройки элементов, в которых сумма двух элементов равна третьему, подсчитать количество таких троек.
- л) Упорядочить массив ϵ n целых чисел таким образом, чтобы элементы с четными и нечетными значениями чередовались (пока имеются элементы разной четности).
- м) В массиве ϵ n целых чисел найти все элементы, равные квадрату другого элемента массива и составить массив из этих элементов.
- н) \hat{A} множестве из n дизъюнктов найти и исключить все одинаковые дизъюнкты.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4

Язык логики предикатов

Цель занятия: Получить навыки формальной записи рассуждений с использованием языка логики предикатов.

Теоретические сведения

Алфавит языка логики предикатов включает следующие группы символов [1, 3, 5]:

- предметные константы: a, b, c, \dots ;
- предметные переменные: x, y, z, \dots ;
- функциональные символы: f, g, h, \dots ;
- предикатные символы: P, Q, R, \dots ;
- логические связки: $\neg, \&, \vee, \rightarrow, \leftrightarrow$;
- кванторы: \forall, \exists .

Кроме того, для задания порядка операций могут использоваться скобки.

Множество D объектов, о которых ведется рассуждение, называется областью интерпретации языка логики предикатов. Например, областью интерпретации может являться множество всех действительных чисел, множество студентов СПбГЭТУ, либо любое другое множество реальных или мыслимых объектов.

Предметные константы соответствуют конкретным элементам множества D , а предметные переменные могут принимать значения в множестве D .

Функциональные символы соответствуют функциям заданным на области интерпретации. Функциональный символ вместе со списком аргументов образует функциональную форму. Например, если D - множество чисел, то функциональная форма $f(x, y)$ может интерпретироваться как двуместная функция сложения чисел: $x + y$.

Термом является всякая предметная константа, предметная переменная либо функциональная форма. Аргументами функциональной формы могут быть любые термы, например $f(a, x, g(c, z))$.

Всякому предикатному символу соответствует свойство (одноместный предикат) или отношение (n -местный предикат, где $n \geq 2$) на объектах области интерпретации. Предикатная форма (или атом) - это предикатный символ вместе со списком своих аргументов-термов. Например, отношению “больше” на множестве чисел может быть сопоставлена двуместная предикатная форма $P(x, y)$. Поскольку при подстановке предметных констант предикатная форма принимает значение T (true) или F (false), можно считать, что n -местный предикат определяет функцию: $D^n \rightarrow \{F, T\}$.

Понятие формулы в логике предикатов определяется следующим образом:

- всякий атом есть формула;
- если A и B - формулы, то $\neg A, A \& B, A \vee B, A \rightarrow B$ и $A \leftrightarrow B$ также формулы;
- если A - формула и x - переменная, то $\forall x A$ и $\exists x A$ - формулы;
- других формул нет.

Квантор всеобщности \forall соответствует словосочетанию “для всех”, т.е. формула вида $\forall x P(x)$ интерпретируется как высказывание: “Для всех объектов области интерпретации выполняется свойство P ”. Квантор \exists существования соот-

ветствует слову “существует”. Например, формула вида $\exists x \exists y Q(x, y)$ интерпретируется как высказывание: “Существует пара объектов в области интерпретации, которые находятся в отношении Q ”.

Квантор вместе с переменной называется квантификацией. **Область действия** некоторой **квантификации** есть формула, к которой применяется эта квантификация. Например в формуле $\exists y (\exists z (P(y, z) \& Q(z)) \rightarrow \forall u R(u, y))$ следующие области действия:

$\exists y$ — $(\exists z (P(y, z) \& Q(z)) \rightarrow \forall u R(u, y))$;

$\exists z$ — $(P(y, z) \& Q(z))$;

$\forall u$ — $R(u, y)$.

Вхождение переменной в формулу называется **связанным**, если оно находится в области действия квантификации по этой переменной или является вхождением в эту квантификацию. Вхождение переменной в формулу называется **свободным**, если оно не является связанным. Переменная может иметь в формуле одновременно свободные и связанные вхождения. Например, в формуле $\forall x (P(x, y) \& \exists y Q(y, x))$ переменная x имеет только три связанных вхождения, а переменная y имеет одно свободное вхождение (в предикате $P(x, y)$), и два связанных (в квантификаторе $\exists y$ и в предикате $Q(y, x)$).

Для того, чтобы записать некоторое утверждение на языке логики предикатов необходимо:

- зафиксировать множество объектов, о которых идет речь, как область интерпретации;
- выделить функциональные связи и отношения (свойства), упоминаемые в данном утверждении и сопоставить им функциональные и предикатные символы соответствующей местности;
- определить логическую структуру утверждения, включая области действия кванторов, и записать утверждение в виде формулы.

Пример. Записать на языке логики предикатов следующее утверждение: “Для любых двух действительных чисел существует третье число, равное разности двух первых”

Решение. Областью интерпретации является множество действительных чисел. Введем двуместные функциональную и предикатную формы для обозначения соответственно разности чисел и отношения равенства:

$f(x, y)$ - “ $x - y$ ”;

$P(x, y)$ - “ $x = y$ ”.

Окончательно, запишем формулу: $\forall x \forall y \exists z P(f(x, y), z)$.

Упражнения

1. Записать на языке логики предикатов следующие утверждения:

а) Для любых трех чисел, если их сумма - четна, то хотя бы одно из этих чисел - четно.

б) Для любых двух чисел, сумма которых - четна, либо оба слагаемых - четны, либо оба - нечетны.

в) Для любых двух чисел, если их сумма - четна, а произведение - нечетно, то оба числа нечетны.

г) Для любых трех чисел, если их произведение - нечетно, то все три числа - нечетны.

д) Ни одна женщина не является одновременно политиком и домашней хозяйкой.

е) Некоторые женщины одновременно являются юристами и членами конгресса.

ж) Каждый второкурсник прочитал хотя бы одну книгу.

з) Кто-то встретил кого-то, а кто-то так никого и не встретил.

и) Каждое простое число, неравное двум, нечетно.

к) Существуют числа, не имеющие общих делителей, кроме единицы.

л) Две прямые, каждая из которых параллельна третьей прямой, параллельны между собой.

м) Судья Джонс не восхищается ни одним жуликом.

н) Если по крайней мере один ученик решил все задачи, то каждую задачу решил по крайней мере один ученик.

о) В Москве живет женщина, имеющая брата в Петербурге, тогда и только тогда, когда в Петербурге живет мужчина, имеющий сестру в Москве.

2. Указать все подформулы, а также области действия квантификаций, свободные и связанные вхождения всех переменных в следующих формулах:

а) $\exists x[(\neg P(x) \ \& \ Q(x)) \rightarrow \forall y \forall z S(z, x, y)]$;

б) $R(w) \vee \exists z[S(z) \leftrightarrow \forall v \forall w(P(v) \ \& \ \neg S(w))]$;

в) $\forall x[P(x) \ \& \ \exists y(R(y) \ \& \ \neg P(y) \rightarrow \forall z S(x, u, z))]$;

г) $S(t, w) \vee \exists x \forall w[(Q(x, w) \rightarrow P(x)) \rightarrow R(w)]$;

д) $[Q(x, y) \leftrightarrow \exists y \exists z P(z, y)] \rightarrow (R(y) \vee \forall z Q(z))$;

е) $\exists v \exists z P(z, v) \ \& \ \neg Q(v, y) \vee \forall x \exists y (R(x) \rightarrow T(x, y))$;

ж) $\forall x \forall u (\neg S(u, x) \vee R(x, u) \vee Q(t)) \leftrightarrow \neg \exists z \exists t P(x, t, z)$;

з) $(P(x, w) \vee \exists x \forall w (Q(x, w) \ \& \ P(x, z)) \rightarrow R(z, w))$.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5

Преобразование формул логики предикатов в клаузальную форму.

Цель занятия: получить практические навыки преобразования формул исчисления предикатов из стандартной формы в клаузальную.

Теоретические сведения

В логике предикатов, как и в логике высказываний, приведение формул к некоторой нормальной (канонической) форме позволяет упростить алгоритмы проверки их свойств и, в конечном счете, решение проблемы дедукции. Однако в логике предикатов задача приведения к нормальной форме является более сложной вследствие более богатого синтаксиса языка, в частности, из-за наличия

квантификаций, свободных и связанных вхождений одной и той же переменной и др.

Преобразование формул логики предикатов из стандартной формы в клаузуальную выполняется в три этапа:

1. Преобразование в предваренную форму.
2. Получение замкнутой формы и сколемизация.
3. Преобразование матрицы в КНФ.

Предваренной формой в логике предикатов называется такое представление формулы, в котором все квантификации размещаются в начале, а затем следует формула, не содержащая квантификаций. Конечную последовательность квантификаций в начале формулы называют префиксом, а не содержащую квантификаций формулу - матрицей. Таким образом, предваренная форма имеет вид:

$$K_1 K_2 \dots K_n M,$$

где символ K_i означает либо \forall -квантификацию, либо \exists -квантификацию для $i=1, \dots, n$, а M - матрица. Квантификации в префиксе относятся к различным переменным и их порядок, в общем случае, имеет значение.

Для любой формулы логики предикатов существует логически эквивалентная ей предваренная форма. Алгоритм получения предваренной формы для произвольной формулы логики предикатов включает следующие шаги:

1. Исключение связок импликации и эквивалентности.
2. Переименование (если необходимо) связанных переменных таким образом, чтобы никакая переменная не имела бы одновременно свободных и связанных вхождений. Это должно быть выполнено для всех подформул рассматриваемой формулы.

3. Удаление квантификаций, область действия которых не содержит вхождений квантифицированной переменной.

4. Сужение области действия отрицаний и снятие двойных отрицаний. При этом помимо законов де Моргана и инволюции используются следующие тождества:

$$\neg(\forall x A) = \exists x(\neg A);$$

$$\neg(\exists x A) = \forall x(\neg A).$$

5. Перенос всех квантификаций в начало формулы. Для этого используются следующие правила:

$$(\forall x A \ \& \ \forall x B) = \forall x(A \ \& \ B);$$

$$(\forall x A \ \vee \ \forall x B) = \forall x(A \ \vee \ B);$$

$$(\forall x A \ \& \ B) = \forall x(A \ \& \ B), \text{ если формула } B \text{ не содержит } x;$$

$$(\exists x A \ \& \ B) = \exists x(A \ \& \ B), \text{ если формула } B \text{ не содержит } x.$$

$$(\forall x A \ \vee \ B) = \forall x(A \ \vee \ B), \text{ если формула } B \text{ не содержит } x;$$

$$(\exists x A \ \vee \ B) = \exists x(A \ \vee \ B), \text{ если формула } B \text{ не содержит } x.$$

При выполнении этого шага некоторые связанные переменные могут быть переименованы. Например, формула $\exists x P(x) \ \& \ \forall x Q(x)$ будет сначала преобразована в $\exists x P(x) \ \& \ \forall y Q(y)$, после чего применены правила преобразования

Пример. Преобразовать в предваренную форму следующую формулу:

$$\exists y \forall x (\neg Q(x, y) \vee \neg P(y)) \rightarrow (R(v) \& \forall v \exists z \forall w S(v, z))$$

Решение.

1. Искключение связок импликации и эквивалентности:

$$\neg \exists y \forall x (\neg Q(x, y) \vee \neg P(y)) \vee (R(v) \& \forall v \exists z \forall w S(v, z))$$

2. Переименование.

$$\neg \exists y \forall x (\neg Q(x, y) \vee \neg P(y)) \vee (R(v) \& \forall u \exists z \forall w S(u, z))$$

3. Удаление ненужных квантификаций:

$$\neg \exists y \forall x (\neg Q(x, y) \vee \neg P(y)) \vee (R(v) \& \forall u \exists z S(u, z))$$

4. Сужение области действия отрицаний и снятие двойных отрицаний:

$$\forall y (\neg \forall x (\neg Q(x, y) \vee \neg P(y))) \vee (R(v) \& \forall u \exists z S(u, z))$$

$$\forall y \exists x (\neg (\neg Q(x, y) \vee \neg P(y))) \vee (R(v) \& \forall u \exists z S(u, z))$$

$$\forall y \exists x (\neg \neg Q(x, y) \& \neg \neg P(y)) \vee (R(v) \& \forall u \exists z S(u, z))$$

$$\forall y \exists x (Q(x, y) \& P(y)) \vee (R(v) \& \forall u \exists z S(u, z))$$

5. Перенос квантификаций в начало формулы.

$$\forall y [\exists x (Q(x, y) \& P(y)) \vee R(v) \& \forall u \exists z S(u, z)]$$

$$\forall y \exists x [Q(x, y) \& P(y) \vee R(v) \& \forall u \exists z S(u, z)]$$

$$\forall y \exists x \forall u [Q(x, y) \& P(y) \vee R(v) \& \exists z S(u, z)]$$

$$\forall y \exists x \forall u \exists z [Q(x, y) \& P(y) \vee R(v) \& S(u, z)]$$

Получена предваренная форма.

Предваренная форма в общем случае может содержать свободные переменные. Однако, при анализе выполнимости достаточно оперировать только замкнутыми формулами, т.е. формулами не содержащими свободных переменных. Действительно, если A - формула, содержащая свободные переменные x_0, \dots, x_n , которая (после переименования) не содержит ни одного связанного вхождения этих переменных, то замкнутая формула $\exists x_1 \dots \exists x_n A$ выполнима тогда и только тогда, когда выполнима формула A . Например, формула $\forall y \exists x [Q(x, y) \& S(u, z)]$ после преобразования в замкнутую форму примет вид: $\exists u \exists z \forall y \exists x [Q(x, y) \& S(u, z)]$.

Всякой замкнутой формуле A можно поставить в соответствие формулу S_A , не содержащую кванторов существования, такую, что формулы A и S_A либо обе выполнимы, либо обе невыполнимы. Таким образом, проверка невыполнимости формулы A может быть сведена к проверке невыполнимости формулы S_A . Форма S_A называется сколемовской формой. Алгоритм получения сколемовской формы (сколемизация) из замкнутой предваренной формы включает следующие шаги:

1. Сопоставить каждой \exists -квантифицированной переменной список предшествующих ей в префиксе \forall -квантифицированных переменных и некоторый функциональный символ, местность которого равна мощности полученного списка.

2. В матрице формулы заменить каждое вхождение каждой \exists -квантифицированной переменной на терм, полученный путем добавления к соответствующему функциональному символу списка аргументов, сопоставленных этой переменной.

3. Удалить из формулы все \exists -квантификации.

Сколемизируем формулу в предваренной форме, полученную в рассмот-

ренном выше примере:

$$\forall y \exists x \forall u \exists z [Q(x, y) \& P(y) \vee R(v) \& S(u, z)]$$

1. Поставим в соответствие каждой \exists -квантифицированной переменной функциональную форму от предшествующих ей в префиксе \forall -квантифицированных переменных:

$$x - f(y);$$

$$z - g(y, u).$$

2. Подставим в формулу функциональные формы:

$$\forall y \exists x \forall u \exists z [Q(f(y), y) \& P(y) \vee R(v) \& S(u, g(y, u))].$$

3. Удалим \exists -квантификации:

$$\forall y \forall u [Q(f(y), y) \& P(y) \vee R(v) \& S(u, g(y, u))].$$

Получили сколемовскую форму.

Таким образом, сколемовская форма - это замкнутая предваренная форма, префикс которой содержит только \forall -квантификации. Клаузальная форма - это сколемовская форма, матрица которой является КНФ. Преобразование матрицы в КНФ выполняется так же, как и в логике предикатов. Для нашего примера после применения правил дистрибутивности получим следующую клаузальную форму:

$$\forall y \forall u [(Q(f(y), y) \vee R(v)) \& (Q(f(y), y) \vee S(u, g(y, u))) \& (P(y) \vee R(v)) \& (P(y) \vee S(u, g(y, u)))].$$

Поскольку все переменные в клаузальной форме связаны только \forall -квантификациями, то префикс может быть опущен. Окончательно имеем множество дизъюнктов:

$$\{(Q(f(y), y) \vee R(v)), (Q(f(y), y) \vee S(u, g(y, u))), (P(y) \vee R(v)), (P(y) \vee S(u, g(y, u)))\}.$$

Для доказательства невыполнимости этого множества можно использовать метод резолюций.

Упражнения

1. Выполнить сколемизацию следующих формул, представленных в предваренной форме:

а) $\exists x \forall y \forall z \exists u [(P(x, y) \& \neg Q(u)) \rightarrow R(z, x, y)];$

б) $\forall v \forall w \forall t \exists z [(R(w, t) \vee S(r)) \leftrightarrow (P(v) \& \neg S(w))];$

в) $\exists y \forall z \exists u \forall x [Q(u, z) \& R(u) \& \neg P(y, u) \rightarrow S(x, u, z)];$

г) $\forall t \exists x \forall w \exists y [(P(t, w) \& Q(x, y) \rightarrow S(y)) \rightarrow R(x)];$

д) $\forall v \forall u \exists y \exists z [(S(u, y) \leftrightarrow P(z, y)) \rightarrow (T(y) \vee Q(z))];$

е) $\forall x \exists y \exists v \exists z [(Q(z, v) \& \neg P(v, y) \vee S(x)) \rightarrow R(x, v)];$

ж) $\exists z \exists t \forall x \forall u [(\neg R(u, z) \vee T(x, u) \vee P(t)) \leftrightarrow \neg Q(x, t, z)];$

з) $\forall y \forall z \exists x \forall w [(T(x, w) \vee (P(x, y) \& S(x, z)) \rightarrow R(x, w)];$

и) $\exists y \forall z \exists u \forall x [(\neg S(y, x) \vee Q(u)) \rightarrow \neg R(z, x, y)];$

к) $\forall v \forall w \forall t \exists z [(P(w, t) \rightarrow \neg Q(r)) \& (R(v) \vee Q(w))];$

2. Преобразовать следующие формулы в клаузальную форму:

- а) $\forall x(\neg(R(y) \& Q(x)) \rightarrow \forall zP(z, x, y))$;
 б) $\neg Q(w, z) \& \forall z[S(z) \leftrightarrow \forall x\forall y(R(y) \& \neg T(x))]$;
 в) $\exists y[P(y) \& \forall y(S(x, y) \& \neg P(y) \rightarrow \exists zR(u, z))]$;
 г) $R(t, w) \vee \neg\exists x\forall w[\neg(P(w) \vee S(x)) \rightarrow \neg Q(w)]$;
 д) $[P(y) \& \neg\exists y\forall zT(z, y)] \rightarrow \neg(R(z) \& \exists zQ(z))$;
 е) $\neg\exists v\forall zS(z, v) \& R(v, y) \vee \neg\forall x\exists y(P(x) \rightarrow R(x, y))$;
 ж) $\forall x\forall u(\neg S(u, x) \vee R(x, u) \vee Q(t)) \rightarrow \forall z(\neg\forall tP(x, t, z))$;
 з) $Q(x, w) \vee \neg\exists x\forall w(P(x, w) \& Q(x, z)) \vee \neg S(z, w)$.

3. Записать следующие предложения в виде формул логики предикатов и преобразовать их в клаузальную форму:

а) Погода плохая тогда и только тогда, когда неверно, что светит солнце и жарко.

б) Если существуют марсиане и они разумны, тогда существуют марсианские растения, которые съедобны и питательны.

в) Если для любых трех чисел одно из них равно сумме двух других, то существуют два числа, одно из которых является квадратом другого.

г) Любой студент любит логику или философию тогда и только тогда, когда существует преподаватель, который любит и логику и философию.

д) Если некоторые прикладные программы несовместимы с некоторыми операционными системами, то все студенты, занимающиеся программированием, изучают операционные системы.

е) Если для любых двух человек, не являющихся гражданами одной страны, существует третий человек, не являющийся соотечественником первых двух, то эти три человека не могут быть членами одной национальной сборной по футболу.

ж) Для любых трех человек, играющих в одной баскетбольной команде, либо рост каждого из них более двух метров, либо один из них является разыгрывающим.

з) Если не существует интеллектуальных машин и если каждая идеальная машина является интеллектуальной машиной, то идеальная машина не существует.

и) Если каждый обманщик является преступником и всякий, кто подстрекает преступника является преступником, то существует трус, который подстрекает обманщика.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 6

Метод резолюций в логике предикатов. Унификация.

Цель занятия: Практическое освоение метода резолюций в логике предикатов.

Теоретические сведения

Подстановка σ есть отображение множества V переменных в множество T термов. Подстановка σ задается множеством упорядоченных пар:

$$\sigma = \{(x_1, t_1), \dots, (x_n, t_n)\},$$

где $x_i \neq t_j$. Пусть t - терм, а σ - подстановка, тогда терм $\sigma[t]$ получается одновременной заменой всех вхождений переменных x в t на их образы относительно σ .

Например, пусть $t = h(f(x), y, v)$ и $\sigma = \{(x, a), (y, g(b, z)), (v, w)\}$, тогда $\sigma[t] = h(f(a), g(b, z), w)$.

Композиция двух подстановок σ_1 и σ_2 есть функция $(\sigma_2 \circ \sigma_1)$, определяемая следующим образом:

$$(\sigma_2 \circ \sigma_1)[t] = \sigma_2[\sigma_1[t]].$$

Иначе говоря, композиция представляет собой результат применения σ_2 к термам подстановки σ_1 с последующим добавлением всех пар из σ_2 , содержащих переменные, не входящие в σ_1 . Например, пусть $\sigma_1 = \{(z, g(x, y))\}$ и $\sigma_2 = \{(x, a), (y, b), (v, c), (z, d)\}$, тогда $(\sigma_2 \circ \sigma_1) = \{(z, g(a, b)), (x, a), (y, b), (v, c)\}$.

Терм t_2 называется конкретизацией (частным случаем, примером) терма t_1 , если существует подстановка σ , такая, что $t_2 = \sigma[t_1]$. Терм t вполне конкретизирован, если он не содержит ни одной переменной.

Применение подстановки к литералу означает применение ее ко всем термам - аргументам этого литерала. Результат применения подстановки σ к литералу L обозначим $\sigma[L]$.

Подстановка σ называется унификатором для множества литералов $\{L_1, L_2, \dots, L_k\}$, если имеет место равенство: $\sigma[L_1] = \sigma[L_2] = \dots = \sigma[L_k]$. Множество $\{L_i\}$ литералов называется в этом случае унифицируемым. Таким образом, унификатор порождает общий пример для множества литералов.

Наиболее общим (или простейшим) унификатором (НОУ) для $\{L_i\}$ называется такой унификатор σ_1 , что, если σ_2 - какой-нибудь унификатор для $\{L_i\}$, дающий $\sigma_2\{L_i\}$, то найдется подстановка σ_3 , такая, что $\sigma_2\{L_i\} = (\sigma_3 \circ \sigma_1)\{L_i\}$. Другими словами, НОУ сводим к любому другому унификатору путем композиции с некоторой подстановкой. Поэтому НОУ порождает наименее конкретизированный общий пример.

Рассмотрим алгоритм, который строит НОУ для унифицируемого множества литералов и сообщает о неудаче, если множество не унифицируемо.

1. Положить $k := 0$, $\sigma_k := \varepsilon$ (пустая подстановка). Перейти к п. 2.

2. Если $\sigma_k\{L_i\}$ не является одноэлементным множеством, то перейти к п. 3.

Иначе положить $\text{НОУ} := \sigma_k$ и закончить работу.

3. Каждая из литер в $\sigma_k\{L_i\}$ рассматривается как цепочка символов и выделяются первые термы - аргументы, не являющихся одинаковыми у всех элементов $\sigma_k\{L_i\}$. Эти подвыражения образуют множество рассогласования V_k . Это множество упорядочиваются таким образом, чтобы в начале располагались переменные, а затем - остальные термы. Пусть V_k - первый элемент V_k , а U_k - следующий за ним элемент. Тогда, если V_k - переменная, не входящая в U_k , то принять $\sigma_{k+1} := \{(U_k, V_k)\} \circ \sigma_k$, $k := k+1$, перейти к п. 2. В противном случае окончить работу.

с отрицательным результатом.

Задача. Найти наиболее общий унификатор и соответствующий общий пример для следующего множества литералов или установить, что множество не-унифицируемо:

$$\{L_i\} = \{P(x, z, v,), P(x, f(y), y), P(x, z, b)\}.$$

Решение. В соответствии с приведенным алгоритмом выполним следующие шаги:

1. Положить $k:=0, \sigma_0:=\varepsilon$.

Имеем $\sigma_0 \circ \{L_i\} = \{P(x, z, v,), P(x, f(y), y), P(x, z, b)\}$. Элементы множества различны, поэтому переходим к составлению множества рассогласования.

2. $B_0 = \{z, f(y), z\} = \{z, f(y)\}$, отсюда $\sigma_1 := \{(z, f(y))\} \circ \varepsilon = \{(z, f(y))\}$.

$\sigma_1 \circ \{L_i\} = \{P(x, f(y), \underline{v}), P(x, f(y), \underline{y}), P(x, f(y), \underline{b})\}$ - элементы множества различны.

3. $B_1 = \{v, y, b\}$, $\sigma_2 = \{(v, y)\} \circ \sigma_1 = \{(v, y)\} \circ \{(z, f(y))\} = \{(z, f(y)), (v, y)\}$.

$\sigma_2 \circ \{L_i\} = \{P(x, f(y), \underline{v}), P(x, f(y), \underline{y}), P(x, f(y), \underline{b})\} = \{P(x, f(y), \underline{y}), P(x, f(y), \underline{b})\}$ - элементы множества различны.

4. $B_2 = \{y, b\}$, $\sigma_3 = \{(y, b)\} \circ \sigma_2 = \{(z, f(b)), (v, b), (y, b)\}$.

$\sigma_3 \circ \{L_i\} = \{P(x, f(b), b), P(x, f(b), b), P(x, f(b), b)\} = \{P(x, f(b), b)\}$.

Получили одноэлементное множество, содержащее общий пример для исходного множества литералов, а σ_3 - есть НОУ.

Два литерала унифицируемы, если они построены из одного предикатного символа, примененного к попарно унифицируемым термам. Поэтому задача унификации сводится к унификации множества термов.

Рассмотрим пример доказательства логического следования с использованием метода резолюций в логике предикатов.

Задача. Записать следующее рассуждение на языке логики предикатов и доказать его справедливость, используя метод резолюций:

Посылки: Некоторым нравится Элвис. Некоторые не любят никого, кому нравится Элвис.

Закключение: Некоторых любят не все.

Решение. Введем необходимые формальные обозначения:

$P(x, y)$ - « x нравится y »; a - «Элвис». Тогда данное рассуждение формально запишется следующим образом:

$$\{ \exists x P(x, a), \exists x \forall y (P(y, a) \rightarrow \neg P(x, y)) \} \models \exists z (\neg \forall v P(v, z)).$$

Переносим заключение в левую часть с отрицанием:

$$\{ \exists x P(x, a), \exists x \forall y (P(y, a) \rightarrow \neg P(x, y)), \neg \exists z (\neg \forall v P(v, z)) \}.$$

Преобразование формул в клаузальную форму:

1. $\exists x P(x, a)$. Формула уже в предваренной форме. Сколемизируя, заменяем x на b и получим: $P(b, a)$;

2. $\exists x \forall y (P(y, a) \rightarrow \neg P(x, y))$. Формула в предваренной форме. Преобразуя матрицу в КНФ и заменяя при сколемизации x на c получим:

$$\neg P(y, a) \vee \neg P(c, y);$$

3. $\neg \exists z (\neg \forall v P(v, z)) = \forall z (\neg \neg \forall v P(v, z)) = \forall z \forall v P(v, z)$. Тiãää êëàóçàëüíáü fíðìà есть: $P(v, z)$.

Получили множество дизъюнктов:

1. $P(b, a)$;
2. $\neg P(y, a) \vee \neg P(c, y)$;
3. $P(v, z)$.

Применяя правило резолюций и унификацию получим следующий вывод (в скобках указаны номера родительских дизъюнктов и применяемый унификатор):

4. $\neg P(c, b)$; (1, 2, $\sigma = \{(y, b)\}$)
5. F (3, 4, $\sigma = \{(v, c), (z, b)\}$)

Упражнения

1. Определить наиболее общий унификатор и соответствующий ему общий пример для следующего множества термов или показать, что множество неунифицируемо.

- а) $\{ f(z, g(y, a), h(b)), f(w, g(c, x), h(u)), f(c, g(v, w), h(x)) \}$;
- б) $\{ h(f(a, g(x)), y, b), h(f(v, z), t, u), h(f(w, z), w, s) \}$;
- в) $\{ g(x, h(y, w), f(a)), g(y, h(d, z), f(t)), g(b, h(x, z), f(y)) \}$;
- г) $\{ h(f(a), g(y, z), y), h(x, g(b, u), c) \}$
- д) $\{ f(c, g(a, b), y), f(x, g(u, v), d) \}$
- е) $\{ g(f(b), f(x), a), g(y, v, b) \}$

2. Записать следующее рассуждение на языке логики предикатов и доказать его справедливость, используя метод резолюций.

- а) Посылки: Все люди смертны. Сократ человек.

Заключение: Сократ смертен.

б) Посылки: Ни один человек не является четвероногим. Все женщины - люди.

Заключение: Ни одна женщина не является четвероногой.

в) Посылки: Арт - мальчик, у которого нет автомобиля. Джейн любит только мальчиков, имеющих автомобили.

Заключение: Джейн не любит Арта.

г) Посылки: Ни один первокурсник не любит второкурсников. Все живущие в Васюках - второкурсники.

Заключение: Ни один первокурсник не любит никого из живущих в Васюках.

д) Посылки: Ни один республиканец или демократ не является социалистом. Норман Томас - социалист.

Заключение: Норман Томас не республиканец и не демократ.

е) Посылки: Всякое рациональное число есть действительное число. Существует рациональное число.

Заключение: Существует действительное число.

ж) Посылки: Все рациональные числа являются действительными числами. Некоторые рациональные числа - целые числа.

Заключение: Некоторые действительные числа - целые числа.

з) Посылки: Все первокурсники встречаются со всеми второкурсниками. Ни один первокурсник не встречается ни с одним студентом предпоследнего курса. Существуют второкурсники.

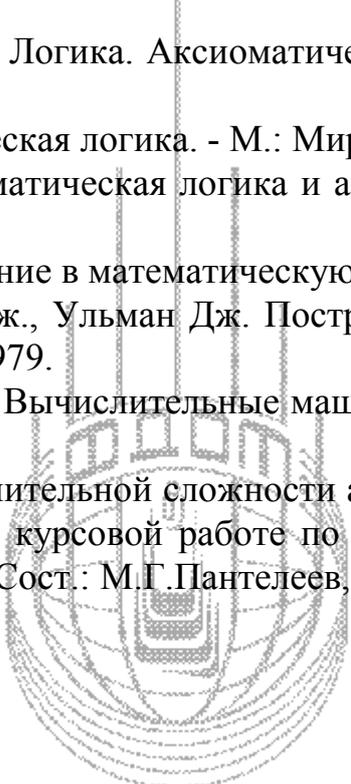
Заключение: Ни один второкурсник не является студентом предпоследнего курса.

и) Посылки: Для любых объектов x , y и z если x есть часть y и y есть часть z , то x есть часть z . Палец есть часть кисти руки. Кисть руки есть часть руки. Рука есть часть человека.

Заключение: Палец есть часть человека.

СПИСОК ЛИТЕРАТУРЫ

1. Тейз А. и др. Логический подход к искусственному интеллекту. - М.: Мир, 1990.
2. Столл Р. Множества. Логика. Аксиоматические теории. - М.: Просвещение, 1968.
3. Клини С. Математическая логика. - М.: Мир, 1973.
4. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. - М.: Наука, 1983.
5. Мендельсон Э. Введение в математическую логику. - М.: Наука, 1971.
6. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. - М.: Мир, 1979.
7. Гэри М., Джонсон Д. Вычислительные машины и трудно решаемые задачи. - М.: Мир, 1982.
8. Исследование вычислительной сложности алгоритмов логического вывода: Методические указания к курсовой работе по дисциплине "Математическая логика и теория алгоритмов"/Сост.: М.Г.Пантелеев, А.С.Календарев; ГЭТУ. СПб., 1997. 32 с.



Факультет
Дистанционных
Образовательных
Технологий