ЛЕКЦИИ ПО ПРЕДМЕТУ «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ» (I СЕМЕСТР)

БАЗОВЫЕ ПОНЯТИЯ ИИ. ОСНОВНЫЕ НАПРАВЛЕНИЯ ИССЛЕДОВАНИЙ В ОБЛАСТИ ИИ

Базовые понятия ИИ

Искусственный интеллект (ИИ) — это область информатики, которая занимается разработкой интеллектуальных компьютерных систем, т.е. систем, обладающих возможностями, которые мы традиционно связываем с человеческим разумом, — понимание языка, обучение, способность рассуждать, решать проблемы и т.д.

Таким образом, основным предметом изучения этой дисциплины являются мыслительные способности человека и способы их реализации техническими средствами.

Искусственный интеллект как научное направление ориентирован на решение определенного типа задач, которые связаны с тонкими и сложными рассуждениями, требующими большой изобретательности и высокой квалификации. Такие задачи называют интеллектуальными. Интеллектуальные задачи относятся к классу трудноформализуемых задач, т.к.:

– задачи не могут быть заданы в числовой форме;

– цели не могут быть выражены в терминах точно определенной целевой функции;

– не существует алгоритмического решения задачи;

– алгоритмическое решение существует, но его нельзя использовать из-за ограниченности ресурсов.

Поэтому представляется совершенно естественным исключить из класса интеллектуальных такие задачи, для которых существуют стандартные методы решения. Примерами таких задач могут служить чисто вычислительные задачи: решение системы линейных алгебраических уравнений, численное интегрирование дифференциальных уравнений и т. д.

Основные направления исследований в области ИИ

1. Принятие решений в задачах диагностики, мониторинга, проектирования и тому подобных задач. Например, диагностика сужения коронарных сосудов, контроль за работой электростанции, синтез электрических цепей и т.д.

2. Разработка естественно-языковых интерфейсов и машинный перевод. Основная сложность – текст можно перевести только на основе понимания его смысла и в конкретном контексте предшествующей информации.

3. Распознавание образов. Это направление включает разработку методов представления информации о зрительных образах. Основной подход — описание классов объектов через определенные значения значимых признаков.

4. Обучение и самообучение. Накопление и формирование знаний на основе анализа и обобщения данных. Простейший метод – корреляционный анализ.

5. Интеллектуальные роботы. Это направление включает в себя вышеперечисленные направления, а также ряд других направлений.

6. Игры и машинное творчество. Это направление включает интеллектуальные игровые задачи — шашки, шахматы и т.п., а также сочинение компьютерной музыки, стихов, сказок, афоризмов и т.п.

Краткая история искусственного интеллекта

Окончательное рождение искусственного интеллекта как научного направления произошло после создания ЭВМ, т.е. в 40-х годах XX века. Вскоре после этого произошло разделение его на два направления: *нейрокибернетика* и *кибернетика «черного ящика».* Эти направления развиваются практически независимо, существенно различаясь как в методологии, так и в технологии. И только в настоящее время стали заметны тенденции к объединению этих частей в единое целое.

Нейрокибернетика

Основную идею этого направления можно сформулировать следующим образом: *Единственный объект, способный мыслить, – это человеческий мозг. Поэтому любое «мыслящее» – устройство должно каким-то образом воспроизводить его структуру.*

Таким образом, нейрокибернетика ориентирована на программно-аппаратное моделирование структур, подобных структуре мозга. Физиологами давно установлено, что основой человеческого мозга является большое количество (до 10^{21}) связанных между собой и взаимодействующих нервных клеток — нейронов. Поэтому усилия нейрокибернетики были сосредоточены на создании элементов, аналогичных нейронам, и их объединении в функционирующие системы. Эти системы принято называть *нейронными сетями,* или *нейросетями.*

Основная область применения нейросетей сегодня — это задачи распознавания образов, например идентификация объектов по результатам аэрофотосъемки из космоса.

Кибернетика «черного ящика»

В основу этого подхода был положен принцип, противоположный нейрокибернетике: *Не имеет значения, как устроено «мыслящее» устройство. Главное, чтобы на заданные входные воздействия оно реагировало так же, как человеческий мозг.*

В рамках этого направления созданы модели и алгоритмы, которые при решении интеллектуальных задач дают результаты сравнимые с результатами, которые получает человек. При этом модели и алгоритмы могут, как воспроизводить процесс принятия решения человеком, так и совершенно отличаться от него.

СИЛЛОГИСТИКА АРИСТОТЕЛЯ

Силлогистика Аристотеля — это первая известная в истории модель дедуктивных рассуждений. Она применялась для ведения научных споров. В ходе такого спора доказательство выдвинутого положения защищалось с помощью ответов двух типов («согласен» или «не согласен») на любые высказывания оппонентов.

Дедуктивные рассуждения — это рассуждения от общего к частному. Суть дедуктивных рассуждений: если общее утверждение верно, то должны быть верны и частные утверждения, определяемые этим общим утверждением.

Примеры дедуктивных рассуждений:

1) Зная, что все предметы падают на землю, можно предполагать, что и подброшенный мяч также упадет.

2) Из двух априорных посылок «Все птицы имеют крылья», «Пингвин — птица» можно сделать заключение о том, что «Пингвин имеет крылья».

3) Из двух априорных посылок «Все птицы имеют крылья», «Все птицы откладывают яйца» можно сделать заключение о том, что «Некоторые существа, откладывающие яйца, имеют крылья».

Последние два примера являются силлогизмами Аристотеля.

Составные части силлогистики

Составными частями силлогистики Аристотеля являются такие понятия как «сущность», «класс» и «квантор».

Сущность — объект, явление, процесс, т.е. то, о чем можно утверждать. Сущности обозначаются маленькими буквами.

Пример: «пингвин», «макака», «орангутанг».

Класс — совокупность (множество) сущностей, объединенных общим именем. Классы могут содержать бесконечное число сущностей, конечное число сущностей, а также быть пустыми. Классы обозначаются большими буквами.

Пример: бесконечный класс — «натуральные числа», конечный класс — «обезьяны», пустой класс — «летающие обезьяны».

Квантор всеобщности. Если его поставить рядом с именем класса, то будет утверждаться нечто, что одновременно истинно для всех сущностей, входящих в класс.

Пример. «Все птицы имеют крылья».

Квантор существования. Если его поставить рядом с именем класса, то в высказывании будет утверждаться нечто, что истинно для какого-то подмножества сущностей, входящих в класс.

Пример. «Некоторые птицы летают».

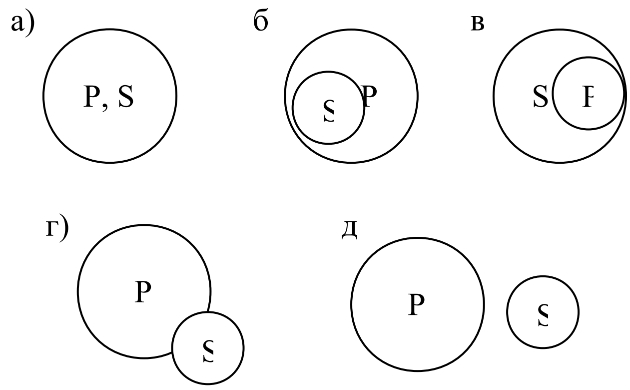
Используя составные части силлогистики Аристотеля, можно задать базовые высказывания силлогистики.

Базовые высказывания силлогистики

Обозначение	Описание	Пример	Классы
Asp	Всякий S есть P; $\forall S \subset P$	«Все обезьяны умные»	S — обезьяны, P — умные существа
Esp	Всякий S не есть P; $\forall S \neg\subset P$	«Все обезьяны не летают»	S — обезьяны, P — летающие существа
Isp	Некоторый S есть P; $\exists S \subset P$	«Существуют хвостатые обезьяны»	S — обезьяны, P — хвостатые существа
Osp	Некоторый S не есть P; $\exists S \neg\subset P$	«Существуют не хвостатые обезьяны»	S — обезьяны, P — хвостатые существа

Здесь S — класс сущностей, о которых что-то утверждается в высказывании, а P определяет, что именно о них говорится.

Смысл базовых высказываний можно наглядно представить с помощью жергоновых отношений.



Базовое высказывание	Варианты жергоновых отношений
Asp	а, б
Esp	д
Isp	а, б, в, г
Osp	в, г, д

Варианты рассуждений

Цель силлогистики — получение правильных рассуждений на основе исходных посылок.

Схема вывода следующая:

Посылка1, Посылка2, …, Посылка N ⇒ Заключение

Посылки и заключение задаются базовыми высказываниями, а знак ⇒ означает, что если истинны все посылки, то истинно и заключение. Предполагается, что в N посылках участвует N+1 класс.

По количеству исходных посылок различают:

выводы ранга 0 — законы силлогистики;

выводы ранга 1 — законы обращения;

выводы ранга 2 — силлогизмы;

выводы ранга 3 и более — сориты.

Законы силлогистики

Законы силлогистики истинны всегда и не зависят от каких-либо посылок.

Закон	Запись	Формулировка	Пример
Закон тождества	$\Rightarrow$Ass	Всякая конкретная сущность, входящая в класс S, обладает всеми свойствами элементов этого класса.	«Осень есть осень»
Закон противоречия	$\Rightarrow\neg$(Asp&Esp)	Всякая сущность не может одновременно входить и не входить в некоторый класс	«Все, кому 70 лет, пожилые люди». «С некоторой уверенностью можно сказать, что все, кому 40 лет, пожилые люди» – некорректное утверждение
Закон исключения третьего	$\Rightarrow$(Isp$\vee$Osp)	Всякая сущность обязательно либо входит, либо не входит в некоторый класс	«Существуют птицы, которые летают, и существуют птицы, которые не летают». «Существуют птицы, которые, достигнув зрелости, перестают летать» – некорректное утверждение

Законы обращения

Законы обращения определяют правила преобразования базовых высказываний.

Закон	Пример
Asp $\Rightarrow$ Isp	Если верно, что «Все птицы откладывают яйца», то верно, что «Некоторые птицы откладывают яйца»
Esp $\Rightarrow$ Eps	Если верно, что «Все обезьяны не летают», то верно, что «Все летающие существа не обезьяны»
Isp $\Rightarrow$ Ips	Если верно, что «Некоторые птицы летают», то верно, что «Некоторые летающие существа являются птицами»
Esp $\Rightarrow$ Ops	Если верно, что «Все обезьяны не летают», то верно, что «Некоторые летающие существа не обезьяны»
Asp $\Rightarrow$ $\neg$Osp	Если верно, что «Все птицы откладывают яйца», то не верно, что «Не-

	которые птицы не откладывают яйца»
$Esp \Rightarrow \neg Isp$	Если верно, что «Все обезьяны не летают», то не верно, что «Некоторые обезьяны летают»

Правильность законов обращения можно проверить с помощью жергоновых отношений.

Силлогизмы

При решении силлогизмов используются так называемые «фигуры силлогизмов», которые определяют расположение классов в посылках и заключении. M – это общий класс в двух посылках.

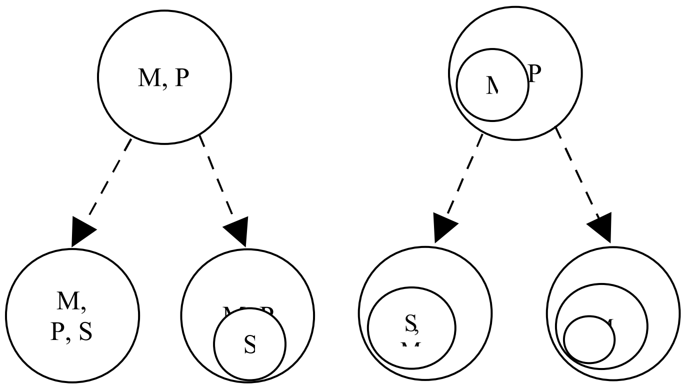
```
1)                              3)
M --------- P                   M --------- P
S --------- M                   M --------- S
S --------- P                   S --------- P

2)                              4)
P --------- M                   P --------- M
S --------- M                   M --------- S
S --------- P                   S --------- P
```

Для каждой фигуры имеется ограниченный набор правильных силлогизмов:

1)	AAA	EAE	EIO	AII	AAI	EAO
2)	EAE	AEE	EIO	AOO	EAO	AEO
3)	AAI	IAI	AII	EAO	OAO	EIO
4)	AAI	AEE	IAI	EAO	EIO	AEO

Правильные силлогизмы можно найти с помощью жергоновых отношений. Например, докажем для 1-й фигуры правильность силлогизм AAA.



Алгоритм решения силлогизма рассмотрим на примере.

Даны посылки:

«Свиньи не летают»

«Свиньи прожорливы»

Нужно получить заключение, зная, что посылки истинны.

Шаг 1. Формализация понятий

Шаг 1.1. Выделить универсум – наиболее общий класс для всех посылок

U = животные

Шаг 1.2. Выделить классы

M – общий класс для двух посылок = «свиньи»

A = «нелетающие животные»

B = «прожорливые животные»

Шаг 2. Представление посылок базовыми высказываниями

Ama

Amb

Шаг 3. Выбор фигуры силлогистики

```
M --------- A
M --------- B
B --------- A
```

3-я фигура

Шаг 4. Выбор правильного силлогизма для фигуры

AAI → Iba

Шаг 5. Словесная интерпретация

«Некоторые прожорливые животные есть нелетающие животные» или «Некоторые прожорливые животные не летают»

Пример решения силлогизмов.

Даны посылки:

«Всякий железный брусок тонет в воде»

«Всякий кирпич не есть железный брусок»

«Всякий кирпич не тонет в воде» — неверно

Шаг 1. U= «объекты», M = «железный брусок», A — «кирпич», B — «объект, который тонет в воде».

Шаг 2.

Amb

Eam

Шаг 3.

```
M --------- B
A --------- M
B --------- A
```

1-я фигура

Шаг 4. Правильного силлогизма для фигуры нет.

Шаг 2. Переставляем посылки:

Eam

Amb

Шаг 3. 4-я фигура

```
A --------- M
M --------- B
B --------- A
```

Шаг 4. EAO

Шаг 5. «Некоторые объекты, которые тонут в воде, не кирпичи»

Замечание 1. Неверные посылки приводят к неверным заключениям. Пример:

«Все птицы летают»

«Пингвин — птица»

«Пингвин летает»

Замечание 2. Не всегда из посылок можно сделать достоверное заключение. Пример:

«Некоторые птицы летают»

«Пингвин — птица»

заключения нет

Расширения силлогистики

Силлогистика может быть расширена за счет двух аспектов:

1. введения отрицательных классов.
2. увеличения числа посылок (более двух).

Введение отрицательных классов

Пример положительного класса: P = «богатые люди». Пример отрицательного класса: ¬P = «небогатые люди».

Введение отрицательных классов делает базовые высказывания более естественными для человека.

Пример:

Пусть у нас имеются классы S = «эскимосы» и P = «богатые люди», тогда высказывание «Все эскимосы (есть) небогатые люди» представляется базовым высказыванием As¬p. Это же высказывание можно записать, не используя отрицательный класс, т.е. «Все эскимосы не являются богатыми людьми» = Esp.

Увеличение числа посылок (более двух)

Если в выводе участвуют боле двух посылок, то такие выводы называются соритами.

Алгоритм решения соритов

На каждом шаге при поиске заключения выбирается пара посылок, которые образуют фигуру силлогизма. Если для такой фигуры найден правильный силлогизм, то выбранная пара порождает заключение.

Если к этому моменту, еще не все посылки исчерпаны, то использованные на данном шаге посылки вычеркиваются, и на их место ставится полученное заключение. Новое множество посылок является исходным для последующего шага, и процесс повторяется.

Замечания.

1. Бывают сориты, имеющие лишнюю посылку. Вывод в них возможен, причем лишняя посылка просто не участвует в выводе.

2. Сориты могут решаться неоднозначно.

3. Если в результате вывода получается несколько заключений, то результирующим является тот, что получен самой длинной цепочкой рассуждений и с возможно большим числом посылок. Остальные считаются тупиковыми.

Пример сорита.

Дано:

1) «Животные, которые не брыкаются, всегда невозмутимы»

2) «У ослов нет рогов»

3) «Буйвол может перебросить вас через ограду»

4) «Животных, которые не брыкаются, нелегко проглотить»

5) «Животное, у которого нет рогов, не может перебросить вас через ограду»

6) «Все животные, кроме буйвола, легко приходят в ярость»

Универсум: животные

Классы: X1= животные, которые не брыкаются, X2= всегда невозмутимые животные, X3= ослы, X4= рогатые животные, X5= буйволы, X6= животные, что могут перебросить через ограду, X7= животные, которых нелегко проглотить.

Базовые высказывания:

1) Ax1x2, 2) Ex3x4, 3) Ax5x6, 4) Ax1x7, 5) E¬x4x6, 6) Ax5x2.

Вывод:

E¬x4x6 ⇒ E¬x6x4 ⇒ Ax6x4

1) Ax1x2, 2) Ex3x4, 3) Ax5x6, 4) Ax1x7, 5) Ax6x4, 6) Ax5x2

1+4 ⇒ 3 фигура ⇒ 7) Ix7x2 ⇒ Ix2x7

2+5 ⇒ 2 фигура ⇒ 8) Ex6x3 ⇒ Ex3x6, Ox3x6

3+6 ⇒ 3 фигура ⇒ 9) Ix2x6 ⇒ Ix6x2

8+9 ⇒ 1 фигура ⇒ 10) Ox2x3

~~7+10 ⇒ 2 фигура ⇒ 11) Ox3x7~~

~~Результат: Ox3x7 = «Некоторые ослы не те животные, которых нелегко проглотить» или «Некоторых ослов легко проглотить».~~

Результат: Ox2x3 = «Некоторые всегда невозмутимые животные не ослы».

Задание. Решить сорит.

Существуют декларативные языки программирования.

Существуют императивные языки программирования.

Все алгоритмические языки программирования являются императивными.

Язык Си является алгоритмическим языком.

ФОРМАЛЬНЫЕ СИСТЕМЫ

Понятие формальной системы

Появление *формальных систем* было обусловлено осознанием того факта, что совершенно различные системы, будь то технические, социальные, экономические или биологические, обладают глубоким сходством.

Формальная система — это совокупность следующих компонентов:

$$Ф = < T, L, Q, R>,$$

где T — множество базовых элементов (конечный алфавит), единственное требование к элементам множества T — для всякого элемента за конечное число шагов можно узнать, принадлежит ли он T или нет, и отличать одни элементы от других, отождествляя одинаковые элементы;

L — множество синтаксических правил построения слов и формул;

Q — множество выделенных синтаксически правильных образований (аксиом), т. е. Q — априорно выведенные формулы;

R — совокупность процедур (правил) вывода одних формул из других.

Формальная система обладает свойством автономности, т.е. если задать ФС, то она самостоятельно начнет генерировать множество выводимых синтаксически правильных совокупностей (теорем). Они будут порождаться в результате применения правил вывода R к совокупностям из множества Q.

Пример. Математика как формальная система.

Алфавит - T: числа, символы переменных, знаки операций и т.д.

Имеются правила построения формул - L. В соответствии с этими правилами выражение `x/5` – является верным, а `x+*6` – неверным.

Аксиомы - Q: например, `3*3 = 9`, `0! = 1`.

Имеются правила преобразования формул - R: например, `(a + b)*c` ⇒ `a*c + b*c`.

Пример. Контекстно-независимая грамматика как формальная система.

Алфавит T: a, b, .

Синтаксические правила L: формулой является символ или последовательность символов, или .

Аксиомы Q: a a.

Правила вывода R: $c_1 \; c_2 \Rightarrow bc_1 \; c_2b$.

Принято, что в этом правиле вывода символы c_1 и c_2 означают какие-то последовательности символов a или b. Символы c_1 и c_2 не являются символами данной формальной системы, они служат посредниками для формализации правил вывода. Здесь a и b называют константами, a – оператором.

Из определения данной формальной системы непосредственно вытекает следующий способ получения допустимых формул:

a a;

ba ab;

bba abb;

bbba abbb и т.д.

Пример. Силлогистика Аристотеля как формальная система.

Алфавит T: буквы, символизирующие имена классов и конкретных сущностей; символы A, E, I, O.

Синтаксические правила L: правила образования нормальных форм высказываний Asp, Esp, Isp, Osp.

Аксиомы Q:

- законы силлогистики: закон тождества (Ass), закон противоречия ($\neg Asp \wedge Esp$), закон исключения третьего ($Isp \vee Osp$).
- некоторое число высказываний, принятых за априорно выведенные, которые берутся из области решаемой задачи.
- Правила вывода R:
- законы обращения (например, $Esp \Rightarrow Eps$).
- набор правильных силлогизмов для фигур силлогистики.
- алгоритм решения соритов.

Таким образом, в формальной системе (ФС), оперирующей теми или иными символами, эти символы воспринимаются просто как элементы, с которыми обращаются согласно определенным правилам, зависящим только от формы выражений, образованных из символов. Понятие истинности появляется только в связи с возможными приложениями (интерпретациями) этой системы.

Доказательство (разрешимость) формальной системы

ФС можно использовать для решения двух задач:

- для получения всех синтаксически верных формул — теорем (теорема — это выводимая синтаксически верная формула);
- для доказательства того, что некоторая формула выводима из заданных аксиом.

Формальная система, позволяющая решать эти задачи, называется разрешимой. Однако определить, является ли формальная система разрешимой, удается не всегда. Причина этого состоит в следующем. Если после какого-то количества правил, примененных последовательно ко всем объектам формальной системы, мы не определили, является ли формула теоремой, то невозможно определить причину этого: то ли рассматриваемая формула не является теоремой, то ли сама формальная система не является разрешимой.

Пример. Разрешимость силлогизмов.

При решении силлогизмов используется четкий алгоритм вывода и возможно получение всего двух теорем, поэтому всегда можно решить первую и вторую задачи.

Интерпретация формальной системы

Интерпретация представляет собой распространение исходных положений какой-либо формальной системы на реальный мир. Интерпретация устанавливает взаимно однозначное соответствие между символами формальной системы и реальными объектами. Только после интерпретации теоремы можно делать выводы о ее истинности или ложности.

Пример. Решение задачи математиком.

Вначале математик изучает реальность, конструируя некоторое абстрактное представление о ней, т. е. некую формальную систему. Затем он доказывает теоремы этой формальной системы. Вся польза и удобство формальных систем как раз и заключаются в их абстрагировании от конкретной реальности. Благодаря этому одна и та же формальная система может служить моделью многочисленных различных ситуаций. Наконец, он возвращается к исходной точке всего построения и даёт интерпретацию теорем, полученных при формализации.

Пример. Вывод в силлогистике Аристотеля.

Укрупнено вывод в силлогистике Аристотеля выполняется в 3 шага.

Шаг 1. Формализация посылок, т.е. представление посылок в виде базовых высказываний.

Шаг 2. Получение заключения, представляемого в виде базового высказывания.

Шаг 3. Словесная интерпретация заключения.

Смысл базовых высказываний (формул) определяется на первом и последнем шагах, а собственно формирование заключения осуществляется «бездумно» на основе преобразования одних формул в другие.

Истинность формальной системы

При интерпретации теоремы и нетеоремы формальной системы приобретают истинность или ложность. На самом деле имеются четыре варианта взаимоотношений между доказательством и значением истинности.

Доказательство	Истинность	Пример	Причина противоречия
теорема	истина	Исходя из аксиом «Все птицы откладывают яйца» и «Пингвин – птица», имеется теорема «Пингвины откладывают яйца»	
теорема	ложь	Исходя из аксиом «Все птицы летают» и «Пингвин – птица», имеется теорема «Пингвины летают»	Неверные аксиомы и правила вывода, не отражающие закономерности предметной области
нетеорема	истина	Исходя из аксиом «Все птицы откладывают яйца» и «Пингвин – птица», нельзя доказать, что «Страусы откладывают яйца», хотя это истина	Неполнота аксиом и неверные правила вывода с точки зрения предметной области
нетеорема	ложь	Исходя из аксиом «Все птицы откладывают яйца» и «Пингвин – птица», нельзя доказать, что «Пингвины не откладывают яйца»	

Возможны формальные системы, в которых аксиомы порождают ложные теоремы, т.е. с синтаксической точки зрения все верно, а с семантической нет. Возможна ситуация и наоборот.

Главная цель построения формальной системы — построить такую формальную систему, в которой теоремы всегда истинны, а нетеоремы всегда ложны, и имеется эффективный алгоритм разрешимости формальной системы.

Ограничения формальных систем

Применение формальных систем на практике имеет ряд ограничений, которые сформулированы в виде самых общих теорем.

Теорема Геделя: Возможны формальные системы, в которых существуют формулы m, такие что ни m ни $\neg m$ не являются доказуемыми.

Теорема Тарского: Возможны формальные системы, в которых во всякой интерпретации найдутся выражения истинные, но недоказуемые.

Теорема Черча: Возможны формальные системы, в которых не существует алгоритма, чтобы отличить теоремы от нетеорем.

ИСЧИСЛЕНИЕ ВЫСКАЗЫВАНИЙ

Исчисление высказываний как формальная система

Силлогистика Аристотеля является одной из наиболее простых формальных систем, что не позволяет задавать сложные отношения предметной области. Например, в силлогистике Аристотеля отсутствуют логические операции. Исчисление высказываний является более сложной формальной системой, а следовательно позволяет задавать более сложные отношения предметной области. Определим компоненты этой формальной системы.

Словарь T:

– строчные буквы (a, b, c и т.д.), обозначающие элементарные высказывания;

– символы логических операций: $\neg$ (отрицание), конъюнкция ($\wedge$), дизъюнкция ($\vee$), импликация ($\rightarrow$), эквивалентность ($\leftrightarrow$);

– скобки «(» и «)».

Правила построения синтаксически правильных формул L:

1. Всякое элементарное высказывание является правильной формулой.

2. Если c_1 и c_2 являются правильными формулами, то правильными формулами являются также $\neg c_1$, $(c_1 \wedge c_2)$, $(c_1 \vee c_2)$, $(c_1 \rightarrow c_2)$, $(c_1 \leftrightarrow c_2)$.

3. Других правильных формул в исчислении высказываний нет.

Аксиомы Q:

Аксиомами являются исходные общезначимые формулы, применимые к решаемой задаче. Для построения аксиом необходимо знать какой смысл вкладывается в такие понятия как «элементарное высказывание», «логическая операция» или «класс формулы».

Элементарные высказывания

Высказывание – это предложение, принимающее только два значения: истина или ложь. Элементарными называются высказывания, которые нельзя разделить на части.

Пример.

Элементарные высказывания: a – «Свиньи не летают» (высказывание всегда истинно), b – «Свиньи прожорливы» (высказывание всегда истинно), c – «Все птицы летают» (высказывание всегда ложно), d – «животное есть птица» (истинность высказывания определяется в зависимости от того, какое животное имеется в виду).

Однако, в другом контексте высказывание «Свиньи не летают» может рассматриваться как сложное высказывание и будет разделено на два высказывания: a – «животное есть свинья», b – «животное не летает».

В общем случае, что является элементарным высказыванием, зависит от целей решаемой задачи.

Логические операции

Используя логические операции, можно составлять сложные высказывания (формулы). Смысл операций представлен таблицей истинности.

a	¬a	b	a∧b	a∨b	a→b	a↔b
T	F	T	T	T	T	T
T	F	F	F	F	F	F
F	T	T	F	T	T	F
F	T	F	F	T	T	T

Приоритет операций в порядке убывания

$\neg \quad \wedge \quad \vee \quad \rightarrow \quad \leftrightarrow$

Пример.

a = «животное есть птица»

b = «животное имеет крылья»

c = «животное откладывает яйца»

$a \rightarrow c$ ($c \rightarrow a$ неверно)

$c \wedge b \rightarrow a$ и $a \rightarrow c \wedge b \Rightarrow c \wedge b \leftrightarrow a$

Классы формул

Класс формулы определяет, на каком количестве интерпретаций формула истинна. Выделяют следующие классы формул.

Формула выполнима, если существует ее интерпретация со значением «истина».

Формула невыполнима, если все ее интерпретации имеют значение «ложь».

Формула общезначима (тождественно истинна), если все ее интерпретации имеют значение «истина».

Формула необщезначима, если не все ее интерпретации имеют значение «истина».

Пример общезначимых формул.

$\neg a \vee a$ – формула истинна при любых интерпретациях.

$a \rightarrow b$ – формула истинна, если a= «животное есть свинья» и b= «животное не летает».

Таким образом, аксиомы составляют общезначимые формулы.

Правила вывода R:

1. Правило подстановки. Согласно ему в формулу, которая уже выведена, можно вместо некоторого высказывания подставить любое другое при непременном условии, что эта подстановка сделана во всех местах вхождения заменяемого высказывания в данную формулу. Такая подстановка сохраняет свойство формулы быть общезначимой.

Пример.

Если в аксиому $(a \rightarrow (a \vee b))$ вместо a подставить любую формулу, например $(b \wedge g)$, то формула $((b \wedge g) \rightarrow ((b \wedge g) \vee b))$ останется общезначимой, что легко доказывается перебором всех комбинаций истинностных значений b и g и проверкой того, что для всех них полученная формула остается истинной.

2. Правило модус поненс (лат. modus ponens) или правило заключения выглядит следующим образом: если c_1 и $(c_1 \rightarrow c_2)$ являются истинными формулами, то формула c_2 также истинна.

Пример.

Пусть имеется формула $a \rightarrow b$, где a= «животное есть змея», b= «животное не имеет ног», тогда зная, что a истина, то b также истина. Другими словами, если мы знаем, что животное – змея, то оно не имеет ног.

3. Правила преобразования.

$c_1 \leftrightarrow c_2 \equiv (c_1 \rightarrow c_2) \wedge (c_2 \rightarrow c_1)$, где знак $\equiv$ означает, что преобразование формул можно выполнять в двух направлениях, т.е. $c_1 \leftrightarrow c_2 \Rightarrow (c_1 \rightarrow c_2) \wedge (c_2 \rightarrow c_1)$ и $(c_1 \rightarrow c_2) \wedge (c_2 \rightarrow c_1) \Rightarrow c_1 \leftrightarrow c_2$

$c_1 \rightarrow c_2 \equiv \neg c_1 \vee c_2$

$\neg\neg c_1 \equiv c_1$

$\neg(c_1 \vee c_2) \equiv \neg c_1 \wedge \neg c_2$

$\neg(c_1 \wedge c_2) \equiv \neg c_1 \vee \neg c_2$

$c_1 \vee (c_2 \wedge c_3) \equiv (c_1 \vee c_2) \wedge (c_1 \vee c_3)$

$c_1 \wedge (c_2 \vee c_3) \equiv (c_1 \wedge c_2) \vee (c_1 \wedge c_3)$

Пример.

$(a \rightarrow (b \vee \neg c)) \rightarrow p \Rightarrow (\neg a \vee (b \vee \neg c)) \rightarrow p \Rightarrow (\neg a \vee b \vee \neg c) \rightarrow p \Rightarrow (\neg\neg a \wedge \neg b \wedge \neg\neg c) \vee p \Rightarrow (a \wedge \neg b \wedge c) \vee p \Rightarrow (a \vee p) \wedge (\neg b \vee p) \wedge (c \vee p)$

Пример: Определение животного из перечня: гепард, тигр, пингвин, альбатрос или крокодил.

Элементарные высказывания:

a_1	животное имеет шерсть
a_2	животное кормит детенышей молоком
a_3	животное есть млекопитающее
a_4	животное имеет перья
a_5	животное летает
a_6	животное откладывает яйца
a_7	животное есть птица
a_8	животное ест мясо
a_9	животное имеет когти
a_{10}	животное имеет острые зубы
a_{11}	животное имеет глаза, направленные вперед
a_{12}	животное есть хищник
a_{13}	животное имеет рыже-коричневый цвет
a_{14}	животное имеет телесные пятна
a_{15}	животное есть гепард
a_{16}	животное имеет черные полосы
a_{17}	животное есть тигр

a_{18}	животное плавает
a_{19}	животное есть пингвин
a_{20}	животное есть альбатрос
a_{21}	животное есть крокодил

Аксиомы:

1) $a_1 \vee a_2 \to a_3$

($a_3 \to a_1 \vee a_2$ не совсем верно)

В более общем случае:

$a_1 \to a_3$

$a_2 \leftrightarrow a_3$

2) $a_4 \vee (a_5 \wedge a_6) \to a_7$

3) $a_8 \vee (a_9 \wedge a_{10} \wedge a_{11}) \to a_{12}$

4) $a_3 \wedge a_{12} \wedge a_{13} \wedge a_{14} \to a_{15}$

5) $a_3 \wedge a_{12} \wedge a_{13} \wedge a_{16} \to a_{17}$

6) $a_7 \wedge a_{18} \to a_{19}$

7) $a_7 \wedge a_5 \to a_{20}$

8) $a_{12} \wedge \neg a_7 \wedge \neg a_3 \to a_{21}$

9)-12) a_1, a_8, a_{13}, a_{16}

Доказать выводимость теоремы a_{17}.

Для доказательства достаточно использовать правило заключения (модус поненс).

Доказательство выводимости формул

В исчислении высказываний существует два основных метода решения проблемы доказательства: семантический метод и синтаксический метод.

Синтаксический подход к доказательству вывода формул

При синтаксическом способе доказательства сначала записывают посылки и, применяя к ним правила вывода, стараются получить из них другие истинные формулы. Из этих формул и исходных посылок выводят последующие формулы, и процесс продолжают до тех пор, пока не будет получено требуемое заключение. Этот процесс, по сути дела, и является логическим выводом, он часто применяется при доказательстве теорем в математике.

Семантический подход к доказательству вывода формул

Семантический подход базируется на следующей теореме:

Теорема B выводима из формул A1, A2, ..., An тогда и только тогда, когда теорема B истинна, если предполагается истинность формул A1, A2, ..., An. Это можно записать в виде:

$$A1,\ A2,\ ...,\ An \Rightarrow B$$

Знак $\Rightarrow$ в доказательствах и выводах следует считать как «верно, что» или «имеет место, что».

Пример.

Выводимость формулы a_{20} из формул $a_7 \wedge a_5 \rightarrow a_{20}$, a_7 и a_5 можно доказать, если доказать, что формула a_{20} истинна, когда одновременно истинны формулы $a_7 \wedge a_5 \rightarrow a_{20}$, a_7 и a_5.

Описание семантического подхода к доказательству вывода формул

Этот подход заключается в следующем: необходимо перечислить все атомы, в ходящие в формулы A1, A2, ..., An, B, и составить таблицу истинности значений для всевозможных комбинаций значений этих атомов. Затем следует осуществить просмотр полученной таблицы, чтобы проверить, во всех ли ее строках, где формулы A1, A2, ..., An имеют значение «истина», формула B также имеет значение «истина». Этот метод всегда применим, но может оказаться слишком трудоемким: для формул, содержащих K высказываний, метод допускает $2^к$ интерпретаций.

Пример: Определение животного.

a_1	животное имеет шерсть
a_2	животное кормит детенышей молоком
a_3	животное есть млекопитающее

Даны аксиомы:

$A1:\ a_1 \vee a_2 \rightarrow a_3$

$A2:\ a_1$

Доказать выводимость (истинность) формулы

$B:\ a_3$

a_1	a_2	a_3	$a_1 \vee a_2$	A1	A2	B
0	0	0	0	1	0	0
0	0	1	0	1	0	1
0	1	0	1	0	0	0
0	1	1	1	1	0	1
1	0	0	1	0	1	0
1	0	1	1	1	1	1
1	1	0	1	0	1	0
1	1	1	1	1	1	1

Синтаксический подход к доказательству вывода формул. Доказательство методом резолюции

Для сложных областей знаний, содержащих большое количество априорных правильно построенных формул (аксиом), необходимо использовать определенные стратегии доказательства, позволяющие преодолеть хаотичность процесса вывода. Далее рассмотрим одну из стратегий — доказательство методом резолюции.

Доказательство базируется на следующих положениях.

1. Существует теорема: $A1,\ A2,\ \ldots,\ An \Rightarrow B$ тогда и только тогда, когда формула $(A1 \wedge A2 \wedge \ldots \wedge An) \rightarrow B$ общезначима, где $A1,\ A2,\ \ldots,\ An$ и B – формулы.

2. Вместо доказательства общезначимости формулы $(A1 \wedge A2 \wedge \ldots \wedge An) \to B$ можно доказать невыполнимость формулы $A1 \wedge A2 \wedge \ldots \wedge An \wedge \neg B$.

Доказательство:

$\neg\neg((A1 \wedge A2 \wedge \ldots \wedge An) \to B) \Rightarrow \neg\neg(\neg(A1 \wedge A2 \wedge \ldots \wedge An) \vee B) \Rightarrow \neg((A1 \wedge A2 \wedge \ldots \wedge An) \wedge \neg B) \Rightarrow \neg(A1 \wedge A2 \wedge \ldots \wedge An \wedge \neg B)$

3. Правило резолюции: $(c_1 \vee c_2) \wedge (c_3 \vee \neg c_2) \equiv (c_1 \vee c_3)$.

Частные случаи: а) $c_1 \wedge (c_2 \vee \neg c_1) \equiv c_2$, б) $c_1 \wedge \neg c_1 \equiv F$.

Таким образом, для доказательства невыполнимости формулы $A1 \wedge A2 \wedge \ldots \wedge An \wedge \neg B$ необходимо прийти к случаю б).

Алгоритм доказательства методом резолюции:

Шаг 1. Привести формулы $A1,\ A2,\ \ldots\ An,\ \neg B$ к конъюнктивно-нормальной форме (КНФ).

Конъюнктивно-нормальная форма – это конъюнкция конечного числа дизъюнкций, т.е. формула вида $L1 \wedge \ldots \wedge Ln$, где Li – дизъюнкты.

Шаг 2. Сформировать из приведенных формул множество дизъюнктов S.

Шаг 3. Выбрать из множества S два дизъюнкта, содержащих утверждение и отрицание переменной. Применить правило резолюции. Полученный дизъюнкт поместить в множество S.

Шаг 4. Если на предыдущем шаге получен дизъюнкт, отличный от F, то перейти к шагу 3.

Пример:

a_1	животное имеет шерсть
a_2	животное кормит детенышей молоком
a_3	животное есть млекопитающее
a_4	животное имеет перья
a_5	животное летает

a_6	животное откладывает яйца
a_7	животное есть птица
a_{18}	животное плавает
a_{19}	животное есть пингвин

Аксиомы:

A1: $a_1 \vee a_2 \rightarrow a_3$

A2: $a_4 \vee (a_5 \wedge a_6) \rightarrow a_7$

A3: $a_7 \wedge a_{18} \rightarrow a_{19}$

A4 - A6: a_4, a_{18}

Доказать выводимость теоремы B: a_{19}.

Шаг 1. Приведение формул к КНФ.

1) $a_1 \vee a_2 \rightarrow a_3 \Rightarrow (\neg a_1 \vee a_3) \wedge (\neg a_2 \vee a_3)$

2) $a_4 \vee (a_5 \wedge a_6) \rightarrow a_7 \Rightarrow (\neg a_4 \vee a_7) \wedge (\neg a_5 \vee \neg a_6 \vee a_7)$

3) $a_7 \wedge a_{18} \rightarrow a_{19} \Rightarrow \neg a_7 \vee \neg a_{18} \vee a_{19}$

Шаг 2. Множество дизъюнктов S = {

1) $\neg a_1 \vee a_3$

2) $\neg a_2 \vee a_3$

3) $\neg a_4 \vee a_7$

4) $\neg a_5 \vee \neg a_6 \vee a_7$

5) $\neg a_7 \vee \neg a_{18} \vee a_{19}$

6) a_4

7) a_{18}

8) $\neg a_{19}$

}

Шаги 3-4. Получение пустого дизъюнкта.

9)= 5)+8) $\neg a_7 \vee \neg a_{18}$

10)= 7)+9) $\neg a_7$

11)= 3)+10) $\neg a_4$

12)= 6)+11) F

Следовательно, выводимость теоремы B из формул $A1-A6$ доказана.

Приведение формулы к конъюнктивно-нормальной форме:

Шаг 1. Исключить из формулы операции эквивалентности, используя правило преобразования: $c_1 \leftrightarrow c_2 \equiv (c_1 \rightarrow c_2) \wedge (c_2 \rightarrow c_1)$.

Шаг 2. Исключить из формулы операции импликации, используя правило преобразования: $c_1 \rightarrow c_2 \equiv \neg c_1 \vee c_2$.

Шаг 3. Внести операции отрицания внутрь формулы, используя законы де Моргана, и погасить двойные отрицания:

а) $\neg(c_1 \wedge c_2) \equiv (\neg c_1 \vee \neg c_2)$

б) $\neg(c_1 \vee c_2) \equiv (\neg c_1 \wedge \neg c_2)$

в) $\neg\neg c_1 \equiv c_1$

Шаг 4. Применить законы дистрибутивности для получения КНФ:

а) $(c_1 \wedge c_2) \vee c_3 \equiv (c_1 \vee c_3) \wedge (c_2 \vee c_3)$

б) $(c_1 \vee c_2) \wedge c_3 \equiv (c_1 \wedge c_3) \vee (c_2 \wedge c_3)$

Шаг 5. Получить приведенную нормальную форму КНФ путем удаления общезначимых дизъюнктов $c_1 \vee \neg c_1$ и повторяющихся литер $c_1 \wedge c_1$.

Пример:

1) $a_1 \vee a_2 \rightarrow a_3 \Rightarrow \neg(a_1 \vee a_2) \vee a_3 \Rightarrow (\neg a_1 \wedge \neg a_2) \vee a_3 \Rightarrow (\neg a_1 \vee a_3) \wedge (\neg a_2 \vee a_3)$

2) $a_4 \vee (a_5 \wedge a_6) \rightarrow a_7 \Rightarrow \neg(a_4 \vee (a_5 \wedge a_6)) \vee a_7 \Rightarrow (\neg a_4 \wedge \neg(a_5 \wedge a_6)) \vee a_7 \Rightarrow (\neg a_4 \wedge (\neg a_5 \vee \neg a_6)) \vee a_7 \Rightarrow (\neg a_4 \vee a_7) \wedge (\neg a_5 \vee \neg a_6 \vee a_7)$

3) $a_7 \wedge a_{18} \rightarrow a_{19} \Rightarrow \neg(a_7 \wedge a_{18}) \vee a_{19} \Rightarrow (\neg a_7 \vee \neg a_{18}) \vee a_{19} \Rightarrow \neg a_7 \vee \neg a_{18} \vee a_{19}$

ИСЧИСЛЕНИЕ ПРЕДИКАТОВ ПЕРВОГО ПОРЯДКА

Исчисление предикатов первого порядка является формальной системой и является расширением исчисления высказываний.

Отличия исчисления предикатов первого порядка от исчисления высказываний

Исчисление высказываний позволяет формализовать лишь малую часть множества рассуждений, т.к. этот аппарат не позволяет учитывать структуру высказываний, которая существует в естественных языках.

Пример:

Зная следующие высказывания `p` = «Все люди смертны» и `q` = «Сократ – человек», мы не можем доказать, что `r` = «Сократ смертен». Мы можем это указать явно: $(p \wedge q) \rightarrow r$, однако данная формула не будет применима для других людей.

Для устранения этого недостатка необходимо высказывание `Q` разделить на две части: «Сократ» (субъект) и «человек» (свойство субъекта), – и записать в виде функции:

`человек(Сократ).`

Так как субъекты могут меняться, следовательно, константу «Сократ» можно заменить на переменную, например:

`человек(x).`

Такая функция называется предикатом. Предикат – это логическая функция, задающая отношение между константами и переменными. Предикат возвращает либо «истину», либо «ложь».

Кроме того, по сравнению с исчислением высказываний в исчислении предикатов добавляются кванторы, которые указывают область действия переменных.

Квантор общности $\forall$ показывает, что формула справедлива для любого `x`.

Пример: $\forall$`x(человек(x)` $\rightarrow$ `смертен(x))`

Квантор существования $\exists$ показывает, что формула справедлива хотя бы для одного `x`.

Пример: $\exists x$(птица(x) $\to$ летает(x))

Пример:

Пусть имеется предикат любит(x, y), который отражает, что x любит y, тогда следующие формулы будут означать.

$\forall x\ \forall y$ любит(x, y)	все любят всех
$\forall x\ \exists y$ любит(x, y)	у всех есть любимый человек
$\forall y\ \exists x$ любит(x, y)	у всех есть кто-то, кто его любит
$\exists x\ \forall y$ любит(x, y)	есть такой человек, который любит всех
$\exists y\ \forall x$ любит(x, y)	есть такой человек, кого любят все
$\exists x\ \exists y$ любит(x, y)	есть такой человек, который кого-нибудь любит

Переменные, находящиеся в сфере действия кванторов, называются связанными, остальные переменные — свободными.

Пример:

В формуле $\forall x\ Q(x, y, z) \lor P(y)$ переменная x - связанная, y, z — свободные переменные.

Исчисление предикатов как формальная система

Рассмотрим исчисление предикатов как формальную систему.

Словарь — компонента T:

а) счетное множество предметных переменных $x_1, x_2, \dots, x_n \dots$;

б) конечное (может быть и пустое) или счетное множество предметных констант $a_1, a_2, \dots$;

в) конечное (может быть и пустое) или счетное множество функциональных букв $f_1, f_2, \dots, f_k, \dots$;

г) конечное (может быть и пустое) или счетное множество предикатных букв $A_1, A_2, \dots$;

д) символы логических операций $\neg, \to, \land, \lor, \leftrightarrow$;

е) символы кванторов $\forall, \exists$;

ж) скобки `(` `)` и запятая.

Правила построения синтаксически правильных формул — компонента L:

а) всякий атом есть формула. Атом — это константа, переменная, предикат, функция;

б) если F1 и F2 – формулы и x – предметная переменная, то каждое из выражений $\neg$F1, F1 $\rightarrow$ F2, F1 $\wedge$ F2, F1 $\vee$ F2, F1 $\leftrightarrow$ F2, $\forall$x F1, $\exists$x F1 есть формула.

Аксиомы — компонента Q:

Систему аксиом исчисления предикатов составляют система аксиом исчисления высказываний и две дополнительные аксиомы:

а) $\forall$x A(x) $\rightarrow$ A(a).

Аксиома говорит о том, что если предикат A(x) истинен для любых x, то и для некоторого a из того же универсума истинность предиката должна сохраняться.

б) A(a) $\rightarrow$ $\exists$x A(x).

Аксиома говорит о том, что если найдется такое a, что предикат A(a) будет истинным, то верно, что существует x, для которого A(x) истинно.

Правила вывода — компонента R:

К правилам вывода, используемым в исчислении высказываний, добавляются еще три правила.

1) Пусть F1 и F2 — две формулы исчисления предикатов. И пусть в F1 переменная x не входит, а в F2 входит в качестве свободной переменной. Пусть, наконец, формула F1 $\rightarrow$ F2 является выводимой, тогда выводима и формула F1 $\rightarrow$ $\forall$x F2.

2) Если x содержится в качестве свободной переменной в F1 и не содержится в таком виде в F2 и если F1 $\rightarrow$ F2 есть выводимая формула, то $\exists$x F1 $\rightarrow$ F2 также является выводимой.

3) Если F – выводимая формула и в F есть кванторы общности и существования, то любая из связанных ими переменных может быть заменена на другую свя-

занную переменную одновременно во всех областях действий квантора и в самом кванторе. Полученная после этого формула также является выводимой.

Свойства системы исчисления предикатов

1. Исчисление предикатов первого порядка непротиворечиво, т.е. из аксиом нельзя доказать выводимость теоремы и нетеоремы.
2. Всякая теорема является общезначимой формулой (см. разделы «Истинность формальной системы» и «Классы формул»).
3. Всякая общезначимая формула является теоремой (см. разделы «Истинность формальной системы» и «Классы формул»).

Доказательство методом резолюции

Доказательство в исчислении предикатов отличается от доказательства в исчислении высказываний следующими позициями:

1. Все формулы необходимо привести к сколемовской стандартной форме (СНФ).
2. В общем случае требуется процедура унификации при поиске пар дизъюнктов, к которым будет применяться правило резолюции.

Говорят, что формула находится в сколемовской стандартной форме, если она находится в пренексной нормальной форме (ПНФ), и ее префикс содержит только кванторы всеобщности.

Сколемовская нормальная форма наиболее близко соответствует фактам и правилам, записываемым на языке Пролог.

Пример сколемовской стандартной формы:

∀x(человек(x) → смертен(x)) ⇒

∀x(¬человек(x) ∨ смертен(x))   – СНФ

∀x ∀y любит(x, y)   – СНФ

Пренексная нормальная форма (ПНФ)

Говорят, что формула находится в пренексной стандартной форме, если она состоит из префикса – конечной последовательности кванторов, и матрицы – формулы не содержащей кванторов и находящейся в КНФ:

$$F = \#_1 x_1 \#_2 x_2 \ldots \#_n x_n\ M,$$

где # – квантор $\exists$ или $\forall$, M – бескванторная формула.

Пример пренексной стандартной формы:

$\exists x(\text{птица}(x) \rightarrow \text{летает}(x)) \Rightarrow$

$\exists x(\neg\text{птица}(x) \vee \text{летает}(x))$ – ПНФ

$\forall y\ \exists x\ \text{любит}(x, y)$ – ПНФ

Алгоритм получения пренексной нормальной формы:

Шаг 1. Исключить логические связки эквивалентности и импликации. С этой целью многократно (пока это возможно) применяется следующее правило: найти самое левое вхождение связок $\rightarrow$ или $\leftrightarrow$ и сделать замены:

$$F1 \leftrightarrow F2 \equiv (\neg F1 \vee F2) \wedge (F1 \vee \neg F2)$$

$$F1 \rightarrow F2 \equiv \neg F1 \vee F2$$

Шаг 2. Продвинуть знаки отрицания до атомов. Для этого многократно (пока это возможно) делаются замены:

$$\neg\neg F \equiv F$$

$$\neg(F1 \vee F2) \equiv \neg F1 \wedge \neg F2$$

$$\neg(F1 \wedge F2) \equiv \neg F1 \vee \neg F2$$

$$\neg(\forall x\ F(x)) \equiv \exists x\ (\neg F(x))$$

$$\neg(\exists x\ F(x)) \equiv \forall x(\neg F(x))$$

Шаг 3. Переименовать связанные переменные таким образом, чтобы ни одна из переменных не имела одновременно и свободных и связанных вхождений. Многократно (пока это возможно) применяется следующее правило: найти самое левое вхождение переменной такое, что это вхождение связано некоторым квантором, но существует еще одно вхождение этой же переменной; затем сделать замену связанного вхождения на вхождение новой переменной.

Допустимы следующие преобразования:

$\#_1 x\ F1(\ \ldots\ \#_2 x\ F2(\ x\ )\ \ldots) \Rightarrow \#_1 x\ F1(\ \ldots\ \#_2 y\ F2(\ y\ )\ \ldots)$
где # – квантор ∃ или ∀.

$\forall x\ F1\ (x) \vee \exists x\ F2(x) \Rightarrow \forall x\ F1\ (x) \vee \exists y\ F2\ (y)$
$\forall x\ F1\ (x) \wedge \exists x\ F2(x) \Rightarrow \forall x\ F1\ (x) \wedge \exists y\ F2\ (y)$

Примеры:

1) $\forall x(P(x) \vee \exists x\ R(x,\ y)) \Rightarrow \forall x(P(x) \vee \exists z\ R(z,\ y))$
2) $\forall x\ P(x) \vee \exists x\ R(x) \Rightarrow \forall x\ P(x) \vee \exists z\ R(z)$

Шаг 4. Удалить квантификации, область действия которых не содержит вхождений квантифицируемых переменных.

Шаг 5. Вынести кванторы, используя следующие равносильности:

$$\#x(F1(x)) \vee F2 \equiv \#x(F1(x) \vee F2)$$
$$\#x(F1(x)) \wedge F2 \equiv \#x(F1(x) \wedge F2)$$
$$\forall x(F1(x)) \wedge \forall x(F2(x)) \equiv \forall x(F1(x) \wedge F2(x))$$
$$\exists x(F1(x)) \vee \exists x(F2(x)) \equiv \exists x(F1(x) \vee F2(x))$$
$$\#_1 x(F1(x)) \vee \#_2 x(F2(x)) \equiv \#_1 x\ \#_2 y(F1(x) \vee F2(y))$$
$$\#_1 x(F1(x)) \wedge \#_2 x(F2(x)) \equiv \#_1 x\ \#_2 y(F1(x) \wedge F2(y))$$

где # – квантор ∃ или ∀.

Пример :

Привести формулу к ПНФ:

$\forall x(P(x) \leftrightarrow \exists x\ R(x)) \rightarrow \forall y\ Q(y)$

Шаг 1.

$\forall x((\neg P(x) \vee \exists x\ R(x)) \wedge (P(x) \vee \neg\exists x\ R(x))) \rightarrow \forall y\ Q(y)$

$\neg\forall x((\neg P(x) \vee \exists x\ R(x)) \wedge (P(x) \vee \neg\exists x\ R(x))) \vee \forall y\ Q(y)$

Шаг 2.

$\exists x\neg((\neg P(x) \vee \exists x\ R(x)) \wedge (P(x) \vee \neg\exists x\ R(x))) \vee \forall y\ Q(y)$

$\exists x(\neg(\neg P(x) \vee \exists x\ R(x)) \vee \neg(P(x) \vee \neg\exists x\ R(x))) \vee \forall y\ Q(y)$

$\exists x((\neg\neg P(x) \wedge \neg\exists x\ R(x)) \vee (\neg P(x) \wedge \neg\neg\exists x\ R(x))) \vee \forall y\ Q(y)$

$\exists x((P(x) \wedge \forall x\neg R(x)) \vee (\neg P(x) \wedge \exists x\ R(x))) \vee \forall y\ Q(y)$

Шаг 3.

$\exists x((P(x) \wedge \forall z\neg R(z)) \vee (\neg P(x) \wedge \exists v\ R(v))) \vee \forall y\ Q(y)$

Шаг 4. Ни один квантор не удаляется.

Шаг 5.

$\exists x(\forall z(P(x) \wedge \neg R(z)) \vee \exists v(\neg P(x) \wedge R(v))) \vee \forall y\ Q(y)$

$\exists x\forall y((\forall z(P(x) \wedge \neg R(z)) \vee \exists v(\neg P(x) \wedge R(v))) \vee Q(y))$

$\exists x\forall y(\forall z\exists v((P(x) \wedge \neg R(z)) \vee (\neg P(x) \wedge R(v))) \vee Q(y))$

$\exists x\forall y\forall z\exists v((P(x) \wedge \neg R(z)) \vee (\neg P(x) \wedge R(v)) \vee Q(y))$

Алгоритм получения сколемовской стандартной формы

Шаг 1. Формулу представить в ПНФ.

Шаг 2. Привести формулу F к $\forall$-формуле.

Шаг 2.1. Найти наименьший индекс i такой, что $\#_1\#_2 \ldots \#_{i-1}$ – все равны $\forall$, но $\#_i = \exists$.

Шаг 2.2. Если $i = 1$, то вместо x_1 в формулу M подставить константу a_1, отличную от констант, встречающихся в M.

Шаг 2.3. Если $i > 1$, то взять новый $(i-1)$-местный функциональный символ f_i, не встречающийся в M, и заменить x_i на $f_i(x_1, \ldots, x_{i-1})$. Функция $f_i(x_1, \ldots, x_{i-1})$ должна каждому набору переменных $x_1, \ldots, x_{i-1}$ сопоставлять соответствующее значение x_i, т.е.

$$x_i = f_i(x_1, \ldots, x_{i-1})$$

Шаг 2.4. Если такого i нет, то формула F является $\forall$-формулой, иначе перейти к шагу 2.1.

Пример:

Привести формулу к СНФ:

$\exists x\ \forall y\ \forall z\ \exists u\ \forall v\ \exists w\ (P(x, y) \to \neg R(z, u, v) \wedge Q(u, w))$

Шаг 1. Приведем формулу к ПНФ:

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

Шаг 2. Приведем формулу к СНФ:

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

подставляем константу **c** вместо **x**

$\forall y\ \forall z\ \exists u\ \forall v\ \exists w\ ((\neg P(\mathbf{c}, y) \vee \neg R(z, u, v)) \wedge Q(u, w)) \Rightarrow$

подставляем функцию **u = f(y,z)**

$\forall y\ \forall z\ \forall v\ \exists w\ ((\neg P(c,y) \vee \neg R(z, \mathbf{f(y,z)}, v)) \wedge Q(\mathbf{f(y,z)}, w)) \Rightarrow$

подставляем функцию **w = g(y,z,v)**

$\forall y\ \forall z\ \forall v\ ((\neg P(c,y) \vee \neg R(z,f(y,z),v)) \wedge Q(f(y,z), \mathbf{g(y,z,v)}))$

Пример доказательства в исчислении предикатов

Имеются аксиомы:

1) «Некоторые обезьяны любят все фрукты»

2) «Ни одна обезьяна не любит горький перец»

Доказать теорему:

3) «Горький перец не является фруктом»

Выделим предикаты:

$M(x)$ – x есть обезьяна

$F(x)$ – x есть фрукт

$P(x)$ – x есть горький перец

$L(x, y)$ – x любит y

Исходные формулы:

1) $\exists x(M(x) \land \forall y(F(y) \rightarrow L(x, y)))$

2) $\forall x(M(x) \rightarrow \forall y(P(y) \rightarrow \neg L(x, y)))$

3) $\neg(\forall x(P(x) \rightarrow \neg F(x)))$ – теорема берется с отрицанием

Преобразование формул в ПНФ:

1) $\exists x\ (M(x) \land \forall y(F(y) \rightarrow L(x, y))) \Rightarrow$

$\exists x\ (M(x) \land \forall y(\neg F(y) \lor L(x, y))) \Rightarrow$

$\exists x\ \forall y\ (M(x) \land (\neg F(y) \lor L(x, y)))$

2) $\forall x(M(x) \rightarrow \forall y(P(y) \rightarrow \neg L(x, y))) \Rightarrow$

$\forall x(\neg M(x) \lor \forall y(\neg P(y) \lor \neg L(x, y))) \Rightarrow$

$\forall x\ \forall y\ (\neg M(x) \lor \neg P(y) \lor \neg L(x, y))$

3) $\neg(\forall x(P(x) \rightarrow \neg F(x))) \Rightarrow$

$\neg(\forall x(\neg P(x) \lor \neg F(x))) \Rightarrow$

$\exists x\ (\neg\neg P(x) \land \neg\neg F(x)) \Rightarrow$

$\exists x\ (P(x) \land F(x))$

Преобразование формул в $\forall$-формулы:

1) $\forall y\ (M(a) \wedge (\neg F(y) \vee L(a, y)))$

2) $\forall x\ \forall y\ (\neg M(x) \vee \neg P(y) \vee \neg L(x, y))$

3) $P(b) \wedge F(b)$

Получение множества дизъюнктов:

S= {

1) $M(a)$

2) $\neg F(y) \vee L(a, y)$

3) $\neg M(x) \vee \neg P(y) \vee \neg L(x, y)$

4) $P(b)$

5) $F(b)$

}

Получение пустого дизъюнкта:

6)= 5)+2) $L(a, b)$ { $y = b$ } - подстановка

7)= 3)+6) $\neg M(a) \vee \neg P(b)$ { $x = a, y = b$ }

8)= 1)+7) $\neg P(b)$

9)=5)+8) **False**

Пустой дизъюнкт получен, следовательно, теорема доказана.

Задание:

Даны аксиомы:

1) Все небедные и умные люди счастливы: $\forall x(\neg poor(x) \wedge smart(x) \rightarrow happy(x))$
2) Человек, читающий книги, - неглуп: $\forall y\ (read(y) \rightarrow smart(y))$
3) Джон умеет читать и является состоятельным человеком: $read(John) \wedge \neg poor(John)$
4) Счастливые люди живут интересной жизнью: $\forall z\ (happy(z) \rightarrow exciting(z))$

Доказать теорему:

Существует ли человек, живущий интересной жизнью: $\exists w(exciting(w))$