Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Казанский (Приволжский) федеральный университет»

ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ
И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Михайлов В.Ю.

ЛОГИКА ПРЕДИКАТОВ КАК ОСНОВА СОЗДАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

КАЗАНЬ - 2025

УДК 51-74

Учебное пособие публикуется по решению
учебно-методической комиссии Института вычислительной математики
и информационных технологий КФУ
Протокол № от 20 г.

заседания кафедры системного анализа
и информационных технологий
Протокол № от 20 г.

Автор-составитель
к.ф.-м.н. Михайлов В.Ю.

Рецензент
к.ф.-м.н. Пшеничный П.В.

Логика предикатов как основа создания интеллектуальных систем**:** Учебное пособие / Михайлов В.Ю.– Казань: Казанский университет, 2025. – 76 с.

Учебное пособие предназначено для студентов, изучающих курс «Математическая логика и теория алгоритмов», а также для преподавателей, ведущих лекционные и практические занятия по данному курсу.

СОДЕРЖАНИЕ

Введение

Настоящее пособие предназначено для студентов, начинающих своё знакомство с такой обширной и глубокой дисциплиной как математическая логика. Мотивационным посылом начального знакомства с языками и методами математической логики, как правило, является способность после освоения небольшого начального фрагмента успешно решать задачи, которые раньше казались студенту сложными.

По ходу продвижения в изучении материала в сознании студента традиционная задача математики: “заменить вычисления рассуждениями”, согласно работам Лейбница, инвертируется и превращается в задачу математической логики: “заменить рассуждения вычислениями”.

Студенты начальных курсов привыкли, что решение любой содержательной задачи состоит из следующих этапов:

1) Запись на некотором языке знаний, имеющихся у нас, и на основе которых мы будем пытаться искать решение задачи. Как правило, это знания о предметной области задачи и условиях самой задачи (что дано?).
2) Точная запись вопроса задачи и точное понимание структуры объекта, который мы должны предъявить в качестве решения задачи.
3) Нахождение и реализация некоторой последовательности действий по построению промежуточных утверждений и объектов, на основе которых будет строиться искомый объект и обосновываться то, что он является решением исходной задачи.

Заметим, что для реализации первого и второго этапов каждый студент использует свой неформальный язык - некоторую смесь естественного языка, математических терминов и обозначений. Иногда по таким записям студентов бывает трудно понять, какую задачу описал и пытается решить студент.

Реализация же третьего этапа представляет для студента основную сложность и требует от него смекалки и так называемых ‘математических способностей’.

Настоящее пособие пропагандирует подход к решению задач, основанный на использовании логики предикатов.

1) Языком для записи знаний о предметной области и знаний об условиях задачи выбирается язык логики предикатов. Знания о предметной области и условия задачи записываются в виде формул логики предикатов подходящей сигнатуры F1,…,Fk.

2) Вопрос задачи также записывается в виде формулы логики предикатов G.

3) Формулируется стандартная логическая задача, которая служит эквивалентом содержательной задачи. В данном пособии к таким стандартным логическим задачам относятся:

- проверка общезначимости формулы F;
- проверка выполнимости формулы F;
- проверка того, что формула G является логическим следствием формул F1, …, Fk.

Для решения стандартных логических задач применяются методы решения такие, как метод семантических таблиц и метод резолюций, по найденным решениям которых мы можем строить решения исходных содержательных задач.

Таким образом, основными целями пособия является:

1) Помощь студентам в изучении языка логики предикатов и отработки у них навыка записи на нем условий задачи, вопроса задачи и содержательных знаний о предметной области задачи.

2) Помощь студентам в освоении методов решения стандартных логических задач, таких как проверка общезначимости, выполнимости формул и задачи проверки логического следствия. Такими методами, достаточно подробно изложенными в пособии, являются метод семантических таблиц и метод резолюций. При изложении материала в пособии основной упор делается на демонстрации решений задач. Теоремы, доказательства которых содержат теоретический материал, не содержащий описаний методов решения задач, приводятся без доказательств.

Хочется отметить, что автор активно пользовался материалами Захарова В.А. и Подымова В.В., опубликованными на сайте кафедры математической кибернетики ВМК МГУ, за что автор выражает им искреннюю признательность.

Немного о смысле слов. ПРЕДИКАТ - латинский термин (предвидеть, предсказывать), обозначающий член предложения сказуемое. Знания удобно представлять тройками вида

Студент слушает лектора - *Субъект Предикат Объект*

В более общем смысле предикат это свойство, атрибут предмета, отношение между предметами, событиями, явлениями.

Глава 1. Язык логики предикатов

1.1. Алфавит

Базовые символы

1. Предметные константы

Ими обозначаются конкретные (именованные, фиксированные) предметы.
Например: я, 2, π, Солнце, c_1, ...

Const — множество всех констант.

2. Предметные переменные

Ими обозначаются безымянные (не фиксированные) предметы. Они будут записываться привычно: x, y', z_4, ...

Var — множество всех переменных.

Далее это множество полагается счётным и заданным однозначно.

3. Функциональные символы

Ими обозначаются операции над предметами.

Например: +, соседнее, lim, ...

Каждому функциональному символу сопоставляется особое натуральное число — местность (арность, число аргументов).

$f^{(k)}$ — запись функционального символа f с обозначением местности k.

Func — множество всех функциональных символов с сопоставленными им местностями.

4. Предикатные символы.

Ими обозначаются отношения между предметами и свойства предметов
Например: < , является_соседом, красный, ...

При задании языка логики предикатов каждому предикатному символу сопоставляется особое натуральное число — местность (арность, число аргументов).

$P^{(k)}$ — запись предикатного символа P с обозначением местности k.

Pred — множество всех предикатных символов с сопоставленными им местностями.

Сигнатурой алфавита логики предикатов называется тройка
⟨Const, Func, Pred⟩.

Кроме базовых символов в алфавит языка логики предикатов входят:

5. Логические операции

5. 1. Логические связки: $\&$ $\vee$ $\neg$ $\to$

5.2 . Кванторы: Квантор всеобщности (читается «для любого предмета»)- $\forall$
Квантор существования (читается «существует предмет»)- $\exists$

6. Знаки препинания: ( ) ,

Устройство языка логики предикатов однозначно определяется выбором сигнатуры: все символы, не обозначенные в сигнатуре, определены однозначно.

1.2. Синтаксис

Правильно построенные выражения логики предикатов бывают двух типов: формулы и термы.

БНФ, определяющая синтаксис формул логики предикатов:
$t ::= x \mid c \mid f^{(n)}(t_1,t_2,...,t_n)$ (синтаксис термов)
$\phi ::= P^{(k)}(t_1,t_2,...,t_k) \mid$ (синтаксис формул)
$(\phi\&\phi) \mid (\phi\vee\phi) \mid (\phi\to\phi) \mid (\neg\phi) \mid (\forall x\ \phi) \mid (\exists x\ \phi)$,
где:
- ϕ — формула
- $t,t_1,t_2,...,t_n$ — термы:
- $x \in Var$
- $c \in Const$
- $f^{(n)} \in Func$
- $P^{(k)} \in Pred$

<u>Термы</u>
$t ::= x \mid c \mid f^{(n)}(t_1,t_2,...,t_n)$
При помощи термов описываются предметы, получающиеся в результате применения заданных функций (операций) к заданным предметам.

Пример: Пусть $z \in Var$; $1 \in Const$; $+^{(2)}, \cdot^{(2)} \in Func$.
Тогда
1. Предмет, обозначенный переменной z: терм z.
2. Предмет, обозначенный константой 1: терм 1.
3. Предмет, получающийся применением операции + к (1) и (2): терм $+(1, z)$.
4. Предмет, получающийся применением операции $\cdot$ к (1) и (3): терм $\cdot(z, +(1, z))$.
5. Более наглядная инфиксная форма записи терма (4): $z\cdot(1+z)$.

Term —обозначение множества всех термов
(над заданными множествами Var, Const, Func).
$\tilde{x}^n$ — сокращённая запись последовательности «$x_1, \dots, x_n$».

Если t — терм, то:
Var_t — множество всех переменных, входящих в терм t.

$t(\tilde{x}^n)$ — синоним записи t, если $Var_t \subseteq \{x_1, \ldots, x_n\}$.
Терм t — основной, если $Var_t = \varnothing$

Пример: если $x \in Var$; $1, 3 \in Const$ и $+^{(2)}, \cdot^{(2)} \in Func$, то терм $3\cdot(1+3)$ является основным, а терм $3\cdot(1+x)$ таковым не является.

Формулы

$\phi ::= P^{(k)}(t_1,t_2,\ldots,t_k) \mid (\phi \& \phi) \mid (\phi \vee \phi) \mid (\phi \to \phi) \mid (\neg\phi) \mid (\forall x \phi) \mid (\exists x \phi)$

При помощи формул описываются отношения между предметами, строящиеся из «базовых» отношений при помощи логических операций. В некоторых случаях (отношение местности 0) формулой может описываться и высказывание, оцениваемое как истина или ложь.

Пусть $P^{(2)}, R^{(1)} \in Pred$; $f^{(2)} \in Func$; $x, y \in Var$.

Тогда

1) Предложению: 'Предмет y и предмет, получающийся из предметов x и y применением операции f, входят в отношение P' соответствует формула
$P(y, f(x, y))$.

2) Предложение: 'Для любого предмета x верно (1)'
записывается в виде формулы $(\forall x\, P(y, f(x, y)))$.

3) Предложение: 'Если верно (2), то предмет y обладает свойством R'
записывается в виде формулы $((\forall x\, P(y, f(x, y))) \to R(y))$.

4) Предложение: 'Хотя бы для одного предмета y верно (3)'
записывается в виде формулы $(\exists y\, ((\forall x\, P(y, f(x, y))) \to R(y)))$.

Формула атомарна (является атомом), если она имеет вид $P(t_1, t_2, \ldots, t_k)$,
где $P^{(k)} \in Pred$ и $t_1,t_2,\ldots,t_k \in Term$

Остальные формулы называются составными.

Form — множество всех формул (в алфавите с заданной сигнатурой).

Приоритет логических операций (в порядке убывания):
$\forall, \exists, \neg$; затем $\&$; затем $\vee$; затем $\to$.

Как работают приоритеты (пример).

Следующие формулы считаются синтаксически одинаковыми:

$\forall x\, \neg P(x) \& \exists y\, R(x, y) \to \exists x\, (\neg P(x) \vee P(y))$

$\forall x\, (\neg P(x)) \& (\exists y\, R(x, y)) \to \exists x\, ((\neg P(x)) \vee P(y))$

$(\forall x\, (\neg P(x))) \& (\exists y\, R(x, y)) \to (\exists x\, ((\neg P(x)) \vee P(y)))$

$((\forall x\, (\neg P(x))) \& (\exists y\, R(x, y))) \to (\exists x\, ((\neg P(x)) \vee P(y)))$

$(((\forall x\, (\neg P(x))) \& (\exists y\, R(x, y))) \to (\exists x\, ((\neg P(x)) \vee P(y))))$

Квантор связывает ту переменную, которая следует за ним.
Область действия внешнего квантора в формуле $\exists x\phi$ — это подформула ϕ.
Вхождение переменной в область действия связывающего её квантора — связанное вхождение.
Вхождение переменной, не являющееся связанным, —свободное вхождение.
Переменная, имеющая свободное вхождение, — свободная переменная формулы.
Пример: Рассмотрим вхождения переменных в формулу
$\Phi = \exists y(\forall x\neg P(y, f(x,y)) \rightarrow R(x))$.
Все вхождения переменной y связаны квантором $\exists$,
вхождение переменной x в подформулу $\neg P(y, f(x,y))$ связано квантором $\forall$,
но вхождение x в подформулу R(x) не связано никаким квантором и поэтому является свободным вхождением переменной x в Φ.
Будем обозначать Var_ϕ — множество всех свободных переменных формулы ϕ.
Если ϕ — формула, то:
$\phi(\tilde{x}^n)$ — синоним записи ϕ, если $Var_\phi \subseteq \{x_1, \ldots, x_n\}$.
Если $Var_\phi = \varnothing$, то ϕ — называется замкнутой формулой, или высказыванием.
Обозначим CForm — множество всех замкнутых формул (в алфавите с заданной сигнатурой).

1.3. Семантика

Как и в логике высказываний, смысл формуле логики предикатов придаёт интерпретация — «мир, в котором живёт и о котором говорит формула».

Интерпретация состоит из
- предметов, населяющих мир;
- операций над предметами (это смысл функциональных символов);
- отношений, связывающих предметы (это смысл предикатных символов).

Определение. *Интерпретация (сигнатуры ⟨Const, Func, Pred⟩)* —
это система ⟨D, Const, Func, Pred⟩, где:
- D — непустое множество предметов
(область интерпретации, предметная область, универсум);
- Const : Const $\rightarrow$ D — оценка констант;
- Func : Func $\rightarrow \cup_{n\geq 1}(D^n \rightarrow D)$ — оценка функциональных символов;

- Pred : Pred → $\cup_{n\geq 1}(D^n \to \{t, f\})$ — оценка предикатных символов.

c = Const(c) — предмет, сопоставленный константе c.

f = Func(f) : $D^n \to D$ — функция, сопоставленная символу $f^{(n)}$

P = Pred(P) : $D^n \to \{t, f\}$ — предикат, сопоставленный символу $P^{(n)}$

Пример. Рассмотрим формулу
$\phi = \forall x\ (P(x) \to \exists y\ (R(x, y)\ \&\ \neg P(f(y))))$.
Сигнатура формулы ϕ:

Const = $\{c_1, c_2\}$, Func = $\{f^{(1)}\}$, Pred = $\{P^{(1)}, R^{(2)}\}$

I - одна из возможных интерпретаций данной формулы:
предметная область: D = {0, 1, 2} состоит из трех предметов;
оценка констант: $c_1 = 0, c_2 = 1$;
оценка функциональных и предикатных символов f (x), P(x), R(x, y):
f(0)=1, f(1)=2, f(2)=0;
P(0)=t, P(1)=f, P(2)=t;
R(0,0)=t, R(0,1)=t, R(0,2)=f,
R(1,0)=t, R(1,1)=f, R(1,2)=t,
R(2,0)=f, R(2,1)=t, R(2,2)=t.

Семантика термов

Определение. Значение $t(\tilde{x}^n)[\tilde{d}^n]$ терма $t(\tilde{x}^n)$ в интерпретации I
на наборе предметов $d_1, \ldots, d_n$ из области интерпретации — это предмет, задаваемый так:

- для терма-переменной x_i : $x_i[\tilde{d}^n] = d_i$
- для терма-константы c: $c[\tilde{d}^n] = \mathbf{c}$;
- для остальных термов: $f(t_1,\ldots,t_k)[\tilde{d}^n] = \mathbf{f}(t_1[\tilde{d}^n],\ldots,t_k[\tilde{d}^n])$

Семантика формул

Отношение выполнимости формулы $\phi(\tilde{x}^n)$ в интерпретации I на наборе предметов $d_1, \ldots, d_n$ из области интерпретации

$I \models \phi(\tilde{x}^n)[\tilde{d}^n]$) определяется индукцией по сложности формулы так.
- для атомарной формулы:

$I \models P(t_1,\ldots,t_k)[\tilde{d}^n]$
$\Leftrightarrow$

$\mathbf{P}(\mathbf{t_1}[\tilde{d}^n],...,\mathbf{t_k}[\tilde{d}^n]) = t$;

- формула имеет вид отрицания:

$I \models (\neg\phi)[\tilde{d}^n]$

$\Leftrightarrow$

$I \not\models \phi[\tilde{d}^n]$

- формула имеет вид конъюнкции:

$I \models (\phi\&\psi)[\tilde{d}^n]$

$\Leftrightarrow$

$I \models \phi[\tilde{d}^n]$ и $I \models \psi[\tilde{d}^n]$.

- вид дизъюнкции:

$I \models (\phi \vee \psi)[\tilde{d}^n]$

$I \models \phi[\tilde{d}^n]$ или $I \models \psi[\tilde{d}^n]$.

- вид импликации:

$I \models (\phi\rightarrow\psi)[\tilde{d}^n]$

$\Leftrightarrow$

$I \not\models \phi[\tilde{d}^n]$ или $I \models \psi[\tilde{d}^n]$.

- квантор всеобщности:

$I \models (\forall x_0\ \phi(x_0,\tilde{x}^n))[\tilde{d}^n] \Leftrightarrow$

для любого предмета d_0 из области интерпретации верно

$I \models \phi(x_0,\tilde{x}^n)[d_0,\tilde{d}^n]$.

- квантор существования:

$I \models (\exists x_0\ \phi(x_0,\tilde{x}^n))[\tilde{d}^n] \Leftrightarrow$

хотя бы для одного предмета d_0 из области интерпретации верно

$I \models \phi(x_0,\tilde{x}^n)[d_0,\tilde{d}^n]$.

$\phi[x_1/d_1,...,x_n/d_n]$ — синоним записи $\phi(x_1,...,x_n)[d_1,...,d_n]$.

Пример (продолжение).

Снова рассмотрим формулу

$\phi = \forall x\ (P(x) \rightarrow \exists y\ (R(x, y)\ \&\ \neg P(f\ (y))))$.

Сигнатура формулы ϕ:

$Const = \{c_1, c_2\}$, $Func = \{f^{(1)}\}$, $Pred = \{P^{(1)}, R^{(2)}\}$

Рассмотрим I - одну из возможных интерпретаций данной формулы:

предметная область: D = {0, 1, 2} состоит из трех предметов;
оценка констант: $c_1 = 0, c_2 = 1$;
оценка функциональных и предикатных символов $f(x), P(x), R(x, y)$:
f(0)=1, f(1)=2, f(2)=0;
P(0)=t, P(1)=f, P(2)=t;
R(0,0)=t, R(0,1=t, R(0,2)=f,
R(1,0)=t, R(1,1=f, R(1,2)=t,
R(2,0)=f, R(2,1=t, R(2,2)=t.

Проверим соотношение $I \models \phi$. Для вычисления значений кванторвых подформул необходимо вычислять значение соответствующих подформул при всех значениях переменной данного квантора.

$I \models P(x)[2]$ x=2 y=0
$I \not\models R(x, y)[y/0, x/2]$
$I \not\models (R(x, y) \& \neg P(f(y)))[y/0, x/2]$

$I \models P(x)[2]$ x=2 y=1
$I \models P(f(y))[1]$
$I \not\models (\neg P(f(y)))[1]$
$I \not\models (R(x, y) \& \neg P(f(y)))[y/1, x/2]$

$I \models P(x)[2]$ x=2 y=2
$I \models P(f(y))[2]$
$\not\models = (\neg P(f(y)))[2]$
$I \not\models (R(x, y) \& \neg P(f(y)))[y/2, x/2]$

$I \models P(x)[2]$ x=2 y= 0,1,2,
$I \not\models (R(x, y) \& \neg P(f(y)))[y/0, x/2]$

$I \not\models (R(x, y) \& \neg P(f(y)))[y/1, x/2]$
$I \not\models (R(x, y) \& \neg P(f(y)))[y/2, x/2]$

$I \models P(x)[2]$

$I \not\models (\exists y (R(x, y) \& \neg P(f(y))))[2]$

$I \models P(x)[2]$ $\qquad\qquad\qquad\qquad\qquad$ $x=2$

$I \not\models (\exists y\, (R(x, y)\ \&\ \neg P(f\,(y))))[2]$

$I \not\models (P(x) \to \exists y\, (R(x, y)\ \&\ \neg P(f\,(y))))[2]$

Значит, $\quad I \not\models \forall x\, (P(x) \to \exists y\, (R(x, y)\ \&\ \neg P(f\,(y))))$

1.4. Выполнимость, общезначимость логическое следствие

Формула $\phi(\tilde{x}^n)$ выполнима в интерпретации I ($I \Vdash \phi$),
если существует набор предметов $\tilde{d}^n$ из области интерпретации I, такой что $I \models \phi(\tilde{x}^n)[\tilde{d}^n]$.

Формула $\phi(\tilde{x}^n)$ истинна в интерпретации I ($I \models \phi$),
если для любого набора предметов $\tilde{d}^n$ из области интерпретации I верно $I \models \phi(\tilde{x}^n)[\tilde{d}^n]$.

Формула ϕ выполнима (сокращение $\Vdash \phi$),
если существует интерпретация, в которой она выполнима.

Формула ϕ общезначима (сокращение $\models \phi$)
(тождественно истинна; является тавтологией), если она истинна в любой интерпретации.

Про невыполнимую формулу также часто говорят, что она тождественно ложна.

Пример. Рассмотрим формулы:

ϕ: $\forall x\, P(x) \to \exists x\, P(x)$ $\quad$ ψ: $\exists x\, P(x) \to \forall x\, P(x)$ $\quad$ χ: $\forall x\, P(x)\ \&\ \forall x\, \neg P(x)$

Интерпретация I_1: $D = \{d\}$, $P(d) = t$.

Легко вычислить: $I_1 \models \phi$; $I_1 \models \psi$; $I_1 \not\models \chi$.

Интерпретация I_2: $D = \{d_1, d_2\}$, $P(d_1) = t$, $P(d_2) = f$

Вычислим: $I_2 \models \phi$; $I_2 \not\models \psi$; $I_2 \not\models \chi$.

Только что было показано, что

1. формулы ϕ, ψ выполнимы;
2. формулы ψ, χ не общезначимы;

А как доказать общезначимость ϕ и невыполнимость χ?

Модели

Интерпретация I называется *моделью* для предложения ϕ, если $I \models \phi$.

Интерпретация I называется *моделью для множества* предложений Γ ($I \models \Gamma$), если она является моделью для каждого предложения из Γ, и в этом случае говорят, что Γ выполняется в I.

Наряду с «модель для формулы/множества» будем также говорить «модель формулы/множества» (без «для»).
Относительно каждой интерпретации I все предложения делятся на
- выполнимые в I («верные»);
- невыполнимые в I («неверные»).
Относительно каждого предложения ϕ все интерпретации делятся на модели для ϕ (адекватно подходящие под устройство ϕ) и не являющиеся моделями для ϕ (неподходящие).

Предложение ϕ называется *логическим следствием* множества предложений Γ ($\Gamma \models \phi$), если любая модель Γ является моделью ϕ.
Другими словами — если для любой интерпретации I верно $I \models \Gamma \Rightarrow I \models \phi$.
Содержательно — если независимо от смысла символов сигнатуры из справедливости всех утверждений из в Γ, обязательно следует справедливость утверждения ϕ.
Отношение $\models$, используемое в таком смысле, будем называть *отношением логического следования*.
Наряду с $\{\psi_1,...,\psi_n\} \models \phi$ будем также писать $\psi_1,...,\psi_n \models \phi$.
Небольшое пояснение:
- $\forall x\ P(x) \models P(c)$:
если все предметы обладают свойством P, то обязательно предмет, обозначенный символом c, обладает свойством P.
- $P(c) \not\models \forall x\ P(x)$:
если предмет, обозначенный символом c, обладает свойством P, то из этого в общем случае не следует, что все предметы обладают свойством P.
Одна из главных задач (и характерное проявление) интеллектуальной деятельности — это извлечение логических следствий из имеющихся баз знаний.
Эта задача возникает в огромном числе областей «разумной деятельности»: экспертные системы, (автоматическое и ручное) доказательство теорем, формальный анализ программ и др.
Пример. Покажем, что представляют собой логические следствия, на простом показательном примере.
Известно, что:
1. Даша любит Сашу,
2. Саша любит пиво,
3. Паша любит пиво и всех тех, кто любит то же, что и он
Любит ли кто-нибудь Дашу?
Попробуем записать эту задачу на языке логики предикатов.

Начнём с сигнатуры алфавита: в неё войдут
- константы Даша, Саша, Паша, пиво;
- предикатный символ $L^{(2)}$: $L(x, y) \rightleftharpoons$ «икс любит игрека».
Условия задачи переписываются так:
1. Даша любит Сашу $\phi_1 \rightleftharpoons L(\text{Даша}, \text{Саша})$
2. Саша любит пиво. $\phi_2 \rightleftharpoons L(\text{Саша}, \text{пиво})$

3. Паша любит пиво и всех тех, кто любит то же, что и он:

$$\phi_3 \rightleftharpoons L(\text{Паша}, \text{пиво})$$

$$\phi_4 \rightleftharpoons \forall x\, (\exists y\, (L(\text{Паша}, y)\ \&\ L(x, y)) \to L(\text{Паша}, x))$$

Любит ли кто-нибудь Дашу?
Правда ли, что из знаний $\phi_1, \ldots, \phi_4$ необходимо следует знание
$\phi_0 \rightleftharpoons \exists x\, L(x, \text{Даша})$?
В конечном итоге задача переписывается так:
проверить соотношение $\phi_1, \phi_2, \phi_3, \phi_4 \models \phi_0$

Теорема о логическом следствии
Для любого предложения ϕ и любого конечного множества предложений $\Gamma = \{\psi_1, \psi_2, \ldots, \psi_n\}$ справедлива равносильность:
$\Gamma \models \phi \Leftrightarrow\ \models \psi_1 \& \psi_2 \& \ldots \& \psi_n \to \phi$.

Проблема общезначимости формул.
Чтобы уметь извлекать логические следствия и в целом анализировать достоверность утверждений, необходимо понимать законы, связывающие достоверность различных утверждений.
Общезначимые формулы представляют собой один из способов записи таких законов — например, закон вида «если верны утверждения $\psi_1, \ldots, \psi_n$, то верно и ϕ» записывается в виде общезначимой формулы $\psi_1 \& \ldots \& \psi_n \to \phi$.
В связи с этим оказывается важна *проблема общезначимости формул*:
для заданной формулы ϕ проверить её общезначимость: $\models \phi$?
Несколько слов о взаимосвязи свойств общезначимости, выполнимости и невыполнимости формул логики предикатов:
формула $\phi(\tilde{x}^n)$ общезначима $\Leftrightarrow$ формула $\psi(\tilde{x}^n) = \neg\phi(\tilde{x}^n)$ невыполнима;
формула $\phi(\tilde{x}^n)$ общезначима $\Leftrightarrow$ формула $\psi = \forall \tilde{x}^n \phi$ общезначима;
формула $\phi(\tilde{x}^n)$ выполнима $\Leftrightarrow$ формула $\psi = \exists \tilde{x}^n \phi$ выполнима.
Здесь $\forall \tilde{x}^n$ — сокращение для $\forall x_1 \ldots \forall x_n$

$\exists \tilde{x}^n$ — сокращение для $\exists x_1 \ldots \exists x_n$

Глава 2. Подходы к решению задачи проверки ощезначимости

2.1. Перебор интерпретаций

Проверять все интерпретации по очереди? Оценим перспективы этого подхода.

Пример 1. Рассмотрим опять формулу
$\phi = \forall x\ (P(x) \rightarrow \exists y\ (R(x, y)\ \&\ \neg P(f\ (y))))$.

Сигнатура формулы ϕ: Const = $\{c_1, c_2\}$, Func = $\{f^{(1)}\}$, Pred = $\{P^{(1)}, R^{(2)}\}$.

Подсчитаем, сколько различных интерпретации может быть у формулы с данной сигнатурой и областью, состоящей из 3-х элементов {0, 1, 2}:

1) Две константы c_1 и c_2 могут быть оценены 3*3=9 способами.
2) Одноместный функциональный символ $f^{(1)}$ может быть проинтерпретирован 3**(3) = 27 способами.
3) Одноместный предикатный символ может быть проинтерпретирован 2**3=8 способами.
4) Двухместный предикатный символ может быть проинтерпретирован 2**(3**2)=2**9=512 способами.

Итак, получаем, что конечных интерпретаций с областью из 3-х элементов у данной формулы существует
9*27*8*512= 995328.
Причем число возможных интерпретаций для формулы будет астрономически возрастать с увеличением области интерпретации и с усложнением сигнатуры формулы.
Можно сделать вывод, что для содержательных формул и достаточно больших конечных областей интерпретаций перебрать все возможные интерпретации можно лишь теоретически. Для бесконечных областей такой перебор даже теоретически не возможен.

2.2. Сведение кванторных формул к формулам логики высказываний для конечных интерпретаций

Если множество M интерпретации конечно, то проверку общезначимости формулы φ можно осуществлять средствами логики высказываний, предварительно заменив кванторные подформулы соответствующими конечными конъюнкциями (для квантора $\forall$) или конечными дизъюнкциями (для квантора $\exists$) :

$\forall x\ \phi(x) = \wedge\ m\in M\ \phi(x/m)$; $\exists x\ \phi(x) = \vee m\in M\ \phi(x/m)$.

Этим способом мы можем проверить общезначимость формулы в интерпретациях с данным количеством элементов. Но это тоже непросто, т.к. возникающие пропорциональные формулы получаются очень громоздкими.

Продемонстрируем идею этого алгоритма на конкретном примере.

Рассмотрим формулу логики предикатов $\forall y\exists x(\neg P(x,y)\wedge P(y,y))$ и выясним, будет ли она выполнима или общезначима на двухэлементном множестве {a,b}. Напомним, что поскольку на этом множестве высказывание вида $\forall x\ R(x)$ эквивалентно конъюнкции $R(a)\wedge R(b)$, а высказывание вида $\exists xR(x)$ — дизъюнкции $R(a)\vee R(b)$, то данная формула равносильна формуле

$\exists x(\neg P(x,a)\wedge P(a,a))\ \wedge\ \exists x(\neg P(x,b)\wedge P(b,b))$, которая, в свою очередь, равносильна формуле

$[(\neg P(a,a)\wedge P(a,a)) \vee (\neg P(b,a)\wedge P(a,a))] \wedge$

$[(\neg P(a,b)\wedge P(b,b)) \vee (\neg P(bob)\wedge P(b,b))]$.

Одна двухместная предикатная переменная $P(x,y)$ исходной формулы как бы распалась на четыре пропозициональных переменных $P(a,a)$, $P(a,b)$, $P(b,a)$, $P(b,b)$ последней формулы, потому что при подстановке в исходную формулу вместо $P(x,y)$ двухместного предиката, определенного на множестве {a,b}, указанные четыре переменные превратятся в конкретные высказывания (вообще говоря, различные). Так что последняя формула есть по сути дела формула логики высказываний. Чтобы это увидеть совсем отчетливо, обозначим указанные четыре пропозициональные переменные буквами P, R, Q, S соответственно. Тогда полученная формула примет вид:

$[(\neg P\wedge P) \vee (\neg Q\wedge P)] \wedge [(\neg R\wedge S) \vee (\neg S\wedge S)]$.

Составив таблицу истинности данной формулы логики высказываний (или каким-либо другим способом, как это делалось в логике высказываний), легко установить, что формула не является тавтологией, но выполнима: она превратится в истинное высказывание, если вместо P и S подставить истинные высказывания, а вместо Q и R — ложные. Применительно к исходной формуле логики предикатов это означает, что она не общезначима на двухэлементном множестве, но выполнима в нем: она превратится в выполнимый предикат, если вместо предикатной переменной $P(x,y)$ подставить в формулу такой конкретный двухместный предикат, который при одинаковых значениях его предметных переменных x и y превращается в истинные высказывания, а при разных — в ложные.

2.3. Сведение задачи проверки общезначимости к задаче с меньшей областью интерпретации

Попытки свести проверку общезначимости формулы φ на интерпретациях с областью, число элементов в которой равно k, к проверке общезначимости этой формулы на интерпретациях с областями, число элементов в которых меньше k, наталкивается на определенные сложности. Так справедливо

Утверждение. Существует формула φ, истинная в каждой интерпретации, область которой состоит из менее k элементов, но опровержимая в некоторой интерпретации, область которой состоит из k элементов.

В качестве частного случая приведем формулу

$\exists x,y,z[R(x,y) \wedge R(x,z) \wedge R(y,z) \wedge \forall t \neg R(t,t)]$, которая будет выполнима на множестве из трех элементов (выполняющий предикат "$x \neq y$") и не выполнима на множестве из двух элементов.

Отрицание этой формулы можно взять в качестве φ в утверждении при k=3.

Формулой, выполнимой на множестве из четырех элементов и не выполнимой ни на каком множестве из трех элементов являются , например, формула

$\exists x,y,z,u\ [R(x,y) \wedge R(x,z) \wedge R(x,u) \wedge R(y,z) \wedge R(y,u) \wedge R(z,u) \wedge \forall t \neg R(t,t)]$

Отрицание ее можно взять в качестве φ в утверждении при k=4.

Для произвольного k попытайтесь построить подобную формулу самостоятельно.

2.4. Аксиомы бесконечности. Теорема Лёвингейма

Сведение проблемы общезначимости со счетно-бесконечных множеств на конечные, для которых проблема общезначимости решается, очень непросто.

Утверждение. Существует не общезначимое предложение, истинное в любой интерпретации с конечной предметной областью. (Такие предложения часто называют ***аксиомами бесконечности***).

Доказательство. Вот пример такого предложения ф:

$\forall x\ \neg R(x,x) \& \forall x\ \forall y\ \forall z\ (R(x,y) \& R(y,z) \rightarrow R(x,z)) \rightarrow \exists x\ \forall y\ \neg R(x,y)$.

Рассмотрим такую интерпретацию I:

Предметная область — множество всех натуральных чисел. Предикат $R(a,b)=t \Leftrightarrow a<b$. Тогда:

$I \models \forall x\ \neg R(x, x)$, т.к. никакое число не меньше себя;

$I \models \forall x\ \forall y\ \forall z\ (R(x,y) \& R(y,z) \rightarrow R(x,z))$, т.к. если $a < b$ и $b < c$, то обязательно $a < c$;

$I \not\models \exists x \forall y \neg R(x,y)$, т.к. среди натуральных чисел не существует максимального.

Следовательно, $I \not\models \varphi$, и предложение φ не общезначимо.

Покажем теперь, что эта формула будет истинной в любой интерпретации с конечным множеством интерпретации M, на котором определено отношение R(x,y), для которого истинны формулы

$\forall x\ \neg R(x, x)$ (антирефлексивность) и

$\forall x\ \forall y\ \forall z\ (R(x,y)\&R(y,z)\to R(x,z))$ (транзитивность). Возьмем произвольный элемент a0 из M. Если $\forall y\ \neg R(a0, y)$ истинна, то формула φ истинна. Если $\forall y\ \neg R(a0, y)$ ложна, то в M найдется элемент a1 такой что R(a0,a1). Легко заметить, что a1 не совпадает с a0 из-за антирефлексивности R. Повторим это рассуждение. Если $\forall y\ \neg R(a1, y)$ истинна, то формула φ истинна. Если $\forall y\ \neg R(a1, y)$ ложна, то в M найдется элемент a2 такой что R(a1,a2). Легко заметить, что a2 не совпадает ни с a0, ни с a1 из-за транзитивности и антирефлексивности R. Продолжаем процесс, который остановится на некотором шаге k в силу конечности M. Т.о. Мы найдем элемент ak из M такой что $\forall y\ \neg R(ak, y)$ истинна, т.е. формула φ истинна.

С другой стороны, справедлива

Теорема Лёвенгейма. Если формула φ выполнима на каком-нибудь бесконечном множестве, то она выполнима и на счетно-бесконечном множестве.

Она показывает , что при проверке общезначимости (выполнимости) формул достаточно рассматривать интерпретации с не более чем счетными множествами.

Мы видим, что рассмотренные подходы к проблеме проверки общезначимости формул логики предикатов сталкиваются как с техническими, так и с теоретическими проблемами.

2.5. Решение проблемы выполнимости (общезначимости) для формул, содержащих только одноместные предикатные символы и не содержащие функциональные символы

В этом случае проблема сводится к проблеме разрешения выполнимости (общезначимости) формулы на некотором конечном множестве, которая, как установлено выше, имеет эффективное (теоретически) решение. Такое сведение осуществляется на основе следующей теоремы и ее следствия.

Теорема. Если формула логики предикатов, содержащая только одноместные предикатные символы, выполнима, то она выполнима на конечном множестве, содержащем не более 2^n элементов, где *n* — число различных предикатных символов, входящих в рассматриваемую формулу.

Следствие. Если замкнутая формула *F(P1,P2,...,Pn),* в которую входят только одноместные предикатные символы *P1,P2,...,Pn*, тождественно истинна на множестве из 2^n элементов, то она общезначима.

2.6. Задача Порецкого П.С.

Относительно девиц, бывших на некоем бале, известны следующие 14 утверждений:

1. Каждая из девиц была или благовоспитанна, или весела, или молода, или красива;
2. Все не танцующие девицы были некрасивы, каждая из танцующих была или молода, или красива, или благовоспитанна;
3. Когда пожилые девицы образовали отдельный кружок, о каждой из оставшихся можно было сказать, что она или красива, или весела, или благовоспитанна;
4. Если выделить всех девиц немолодых и некрасивых, то останутся лишь благовоспитанные и веселые девицы;
5. Если же выделить всех девиц невеселых, то останутся благовоспитанные, молодые и красивые;
6. Таких девиц, которые, будучи молоды и веселы, не обладали бы вдобавок ни красотой, ни благовоспитанностью, на балу не было;
7. Между молодыми девицами не было таких, которые, обладая красотой и веселостью, были бы не благовоспитанны;
8. Каждая благовоспитанная девица была или молода, или весела, или красива;
9. Все девицы, соединявшие красоту с благовоспитанностью, были одни веселы, другие молоды;
10. Каждой невеселой девице недоставало или молодости, или красоты, или благовоспитанности;
11. Все те веселые девицы, которые, не отличаясь молодостью, обладали благовоспитанностью, были красивы;
12. Немолодые девицы были одни не благовоспитанны, другие не веселы, третьи не красивы;
13. Между некрасивыми девицами не было таких, которые с благовоспитанностью соединяли бы молодость и веселость;
14. Наконец, когда уехали все неблаговоспитанные, невеселые, немолодые и некрасивые девицы, на балу девиц более не осталось.

Вопрос: Возможно ли такое?

Решение.

Введем четыре элементарных предиката

К(х) – х –красива;

В(х) - х – весела;

М(х) – х-молода;

Б(х) – х –благовоспитанна;

Условие 1, записанное в виде формулы логики предикатов:

$$Ф1(х) = (Б(х) \vee В(х) \vee М(х) \vee К(х)),$$

Условие 2, записанное в виде формулы логики предикатов:

$$Ф2(х) = (\neg К(х) \oplus (М(х) \vee К(х) \vee Б(х))),$$

Условие 3, записанное в виде формулы логики предикатов:

$$Ф3(х) = (\neg М(х) \oplus (К(х) \vee В(х) \vee Б(х))),$$

Условие 4. $Ф4(х)= \neg(\neg М(х) \& \neg(К(х)) \rightarrow В(х) \& Б(х)$,

Условие 5. $Ф5(х)= В(х) \rightarrow Б(х) \& М(х) \& К(х)$,

…

Условие 10. $Ф10(x)= \neg В(x) \rightarrow \neg Б(x) \vee \neg М(x) \vee \neg К(x)$,

…

Условие 14. $Ф14(x) = (\neg Б(x) \vee \neg В(x) \vee \neg М(x) \vee \neg К(x))$

Задача состоит в проверке выполнимости формулы
$\varphi = \exists x\, (Ф1(x)\ \&\ Ф2(x)\ \&\ \ldots.\ \&\ Ф14(x))$

Заметим, что все используемые предикаты – одноместные.
Поэтому по теореме в качестве предполагаемого интерпретационного множества достаточно взять
$M=\{b1,b2,b3, \ldots ,b16\}$, где:
для b1 справедливо К(b1)=0, В(b1)=0, М(b1)=0, Б(b1)=0;
для b2 справедливо К(b2)=0, В(b2)=0, М(b2)=0, Б(b2)=1;
для b3 справедливо К(b3)=0, В(b3)=0, М(b3)=1, Б(b3)=0;
…
для b16 справедливо К(b16)=1, В(b16)=1, М(b16)=1, Б(b16)=1.

Решение:
1) Условие 1, $Ф1(x) = (Б(x) \vee В(x) \vee М(x) \vee К(x))$
говорит, из множества М можно исключить элемент b1.
2) Условие 2, $Ф2(x)= \neg К(x) \oplus (М(x) \vee К(x) \vee Б(x)$,
говорит, что из М можно исключить элементы b2,b3,b4, b6,b7,b8.
3) Условие 3, $Ф3(x) = \neg М(x) \oplus (К(x) \vee В(x) \vee Б(x)$,
говорит, что из М можно исключить элементы
b2,b3,b5, b6,b9,b10,b13,b14.
Т.о. осталось проверить могут ли в М для проверки выполнимости использоваться элементы b11, b12, b15, b16.
4) Условие 4, $Ф4(x)= \neg(\neg М(x)\ \&\neg(К(x)) \rightarrow В(x)\ \&\ Б(x)$,
говорит о том, что можно исключить b11.
5) Условие 5, $Ф5(x)= В(x) \rightarrow Б(x)\ \&\ М(x)\ \&\ К(x)$,
говорит о том, что можно исключить b15.
6) Условие 10, $Ф10(x)= \neg В(x) \rightarrow \neg Б(x) \vee \neg М(x) \vee \neg К(x)$,
Говорит о том, что можно исключить b12.
7) Условие 14, $Ф14(x) = \neg Б(x) \vee \neg В(x) \vee \neg М(x) \vee \neg К(x)$
Говорит о том, что можно исключить b16.

Таким образом, в М нет ни одного элемента, удовлетворяющего всем условиям задачи. Следовательно, формула φ не выполнима во всех интерпретациях, у которых множество интерпретации состоит из 16 элементов. Следовательно, φ просто не выполнима. А следовательно, ответ на вопрос в задаче Порецкого: «такого быть не может».

Глава 3. Метод семантических таблиц

3.1. Сведение задачи проверки общезначимости к проверке выполнимости семантических таблиц

Семантическая таблица (логики предикатов) — это упорядоченная пара множеств формул (логики предикатов), такая что хотя бы одно из этих множеств непусто.

Будем называть два множества семантической таблицы её левой частью (первое) и правой частью (второе).

Будем записывать семантическую таблицу так: $\langle \Gamma \mid \Delta \rangle$, где Γ — левая часть и Δ — правая часть.

В записи множеств в семантических таблицах иногда будем опускать фигурные скобки и писать «,» вместо «$\cup$».

Например, $\langle P(c),\ Q(x) \mid \exists x\ Q(x) \rangle$ — это семантическая таблица с левой частью $\{P(c), Q(x)\}$ и правой частью $\{\exists x\ Q(x)\}$

Содержательно, смысл семантической таблицы в интерпретации I состоит в том, что все формулы φ в левой части в интерпретации I принимают значение «истина» $I \models \varphi$, а все формулы в правой части «ложь» $I \not\models \varphi$.

Таблица$T = \langle \Gamma \mid \Delta \rangle$ называется закрытой, если $\Gamma \cap \Delta \neq \varnothing$.

Например,

-таблица $\langle P(c), \forall x\ Q(x) \mid \forall x\ Q(x), R(c) \rangle$ закрыта;

-таблица $\langle P(x), \neg P(x) \mid P(y), Q(x) \rangle$ незакрыта.

Содержательно, если $\phi \in \Gamma \cap \Delta$, то это означает, что получено явное противоречие: соотношения $I \models \phi$ и $I \not\models \phi$

Таблица $T = \langle \Gamma \mid \Delta \rangle$ атомарна, если все формулы из $\Gamma \cup \Delta$ атомарны. Например,

-таблица $\langle P(x) \mid Q(f(c), x), P(d) \rangle$ атомарна

-таблицы $\langle \forall x\ P(x) \mid Q(f(c), x), P(d) \rangle$ и $\langle P(x) \mid Q(f(c), x) \vee P(d) \rangle$ не атомарны

Пусть $\tilde{x}^n$ все свободные переменные формул из $\Gamma \cup \Delta$.

Таблица $T = \langle \Gamma \mid \Delta \rangle$ выполнима, если существуют интерпретация I и набор предметов $\tilde{d}^n$ из области интерпретации, такие что

$I \models \phi(\tilde{x}^n)[\tilde{d}^n]$ для любой формулы ϕ из Γ

$I \not\models \psi(\tilde{x}^n)[\tilde{d}^n]$ для любой формулы ψ из Δ

Содержательно, выполнимость таблицы означает, что в имеющихся соотношениях вида $I \models \phi$ и $I \not\models \psi$ нет противоречия.

Примеры:

1. Семантическая таблица $\langle P(x) \mid Q(f(\mathbf{c}),x)\rangle$ выполнима (и незакрыта, и атомарна). Интерпретация и предметы, подтверждающие выполнимость:
$D=\{d\}$, $P(d)=t$, $Q(f(c),d)=f$, $d_x = d$

2. Семантическая таблица
$T = < \{\exists x\, P(x), \neg P(y)\}\ ;\ \{\forall x P(x), P(x)\ \&\ \neg P(x)\}>$
выполнима. Ее выполнимость подтверждает интерпретация
$I=<D_I, Pred>$:$D_I =\{d_1, d_2\}$, $P(d_1)$=true, $P(d_2)$=false, и набор d_1, d_2 значений свободных переменных x, y.

3. Семантическая таблица
$T = < 0\ ;\ \{\exists y \forall x R(x,y) \rightarrow\ \forall x \exists y R(x,y)\}>$ невыполнима (и незакрыта, и неатомарна).

Теорема (о табличной проверке общезначимости)
Для любой формулы ϕ справедлива равносильность:
$\models \phi \Leftrightarrow$ таблица $\langle \mid \phi\rangle$ невыполнима.
Доказательство.
$\models \phi\ (\tilde{}x^n)$
$\Leftrightarrow$
$I \models \phi(\tilde{}x^n)[\tilde{}d^n]$ для любой интерпретации I и любого набора предметов $\tilde{}d^n$
$\Leftrightarrow$
таблица $\langle \mid \phi\rangle$ невыполнима.
Утверждение. Любая закрытая таблица невыполнима.

Утверждение. Любая незакрытая атомарная таблица выполнима.

Таким образом, для проверки общезначимости формул достаточно придумать правила преобразования таблиц, позволяющие извлекать «явные противоречия» (закрытые таблицы) из таблиц, содержащих «неявные противоречия» (невыполнимых).
Для примера рассмотрим такую невыполнимую таблицу: $\langle \forall x\, P(x) \mid P(c)\rangle$
Чтобы преобразовать эту таблицу в закрытую, достаточно заметить, что если утверждение P(x) выполняется для любого предмета x, то оно выполняется, в частности, и для предмета, обозначенного константой c
Значит, можно подставить на место x константу c и получить выполнимость утверждения P(c).
Добавив это утверждение в левую часть, получим закрытую таблицу:
$\langle \forall x\, P(x), P(c) \mid P(c)\rangle$.

Чтобы строго сформулировать соответствующее правило, следует строго определить, что такое «подставить».

3.2. Подстановки

Пусть заданы множество переменных Var и множество термов Term.
Подстановка — это отображение $\theta : Var \rightarrow Term$.
Область подстановки θ: $Dom_\theta = \{x \mid x \in Var, \theta(x) \neq x\}$.
Подстановка конечна, если её область конечна.
Subst — множество всех конечных подстановок.
$\{x_1/t_1, \ . \ . \ . \ , x_n/t_n\}$ — это конечная подстановка θ, для которой верно: $Dom_\theta = \{x_1,\ldots,x_n\}$, $\theta(x_i) = t_i$, $1 \leq i \leq n$.
Запись x/t, где $x \in Var$ и $t \in Term$, называется связкой.
Содержательно, связка x/t означает, что при применении (выполнении) подстановки переменная x должна быть заменена на терм t.
ε — это тождественная (пустая) подстановка: $Dom_\varepsilon = \varnothing$
Пусть E — логическое выражение логики предикатов (терм или формула) и θ — подстановка
Результат $E\theta$ применения подстановки θ к E определяется так:

$x\theta = \theta(x)$ $\quad (x \in Var)$

$c\theta = c$ $\quad (c \in Const)$

$f(t_1,...,t_n)\theta = f(t_1\theta,...,t_n\theta)$ $\quad (f \in Func, t_1,...,t_n \in Term)$

$P(t_1,...,t_n)\theta = P(t_1\theta,...,t_n\theta)$ $\quad (P \in Pred)$

$\phi\&\psi)\theta = (\phi\theta\&\psi\theta)$ $\quad (\phi,\psi \in Form)$

$(\phi \vee \psi)\theta = (\phi\theta \vee \psi\theta)$

$(\phi\rightarrow\psi)\theta = (\phi\theta\rightarrow\psi\theta)$

$(\neg\phi)\theta = (\neg\phi\theta)$

$(\forall x\ \phi)\theta = (\forall x\ \phi\theta')$ $\quad (\theta'(x) = x;\ \theta'(y) = \theta(y)$, если $y \neq x)$

$\exists x\ \phi)\theta = (\exists x\ \phi\theta')$

Иными словами, $E\theta$ получается из выражения E так:
1. Если E — терм, то все вхождения переменных заменяются на их θ-образы.
2. Если E — формула, то все **свободные** вхождения переменных заменяются на их θ-образы

Пример применения подстановки к формуле:
Пусть $\phi = \forall x(P(x)\rightarrow\neg R(y))\rightarrow R(f(x))\vee\exists y\ P(y)\vee R(u)$ и
$\theta = \{x/g(x, c), y/x, z/f(z)\}$.
Выделяются все свободные вхождения переменных в ϕ

$$\forall x\ (P(x) \rightarrow \neg R(\mathbf{y})) \rightarrow R(f\ (\mathbf{x})) \vee \exists y\ P(y) \vee R(\mathbf{u})$$

Все выделенные вхождения заменяются согласно θ:

$\phi\theta = \forall x\ (P(x) \to \neg R(x)) \to R(f(g(x, c))) \vee \exists y\ P(y) \vee R(u)$.

При применении подстановок для выделения «частных случаев» следует соблюдать осторожность.

Например: пусть дана формула

$\phi(x) = \forall x\ \exists y\ P(x,y) \to \exists y\ P(x,y)$, означающая, например, утверждение

«если у каждого есть дед, то у x тоже есть дед».

Очевидно, что $\models \phi(x)$.

Однако $\phi\{x/y\} = \forall x\ \exists y\ P(x,y) \to \exists y\ P(y,y)$ будет означать

«если у каждого есть дед, то есть и тот, кто сам себе дед» Очевидно, что $\not\models \phi\theta$

Почему смысл формулы после применения подстановки так исказился?

Переменная x свободна для терма t в формуле ϕ, если ни одно свободное вхождение переменной x не лежит в областях действия кванторов, связывающих переменные из Var_t

Подстановка $\theta = \{x_1/t_1, \ldots, x_n/t_n\}$ — правильная для формулы ϕ, если для каждой связки x_i / t_i переменная x_i свободна для терма t_i в формуле ϕ.

Например, для формулы $\forall x\ \exists y\ P(x, y) \to \exists y P(x,y)$ подстановка $\{x/f(u,v)\}$ — правильная: все вхождения u и v в подставляемый терм свободны, а подстановка $\{x/y\}$ — неправильная: вхождение y в подставляемый терм оказывается связанным.

Утверждение (о правильной подстановке).

Для любых формулы $\phi(\tilde{x}^n,x)$, интерпретации I, набора предметов $\tilde{d}^n$ и подстановки $\{x/t(\tilde{x}^n)\}$, правильной для ϕ, верно:

$I \models \phi[\tilde{d}^n, t[\tilde{d}^n]] \Leftrightarrow I \models \phi\{x/t\}[\tilde{d}^n]$

Для доказательства общезначимости формул (и более широко — невыполнимости таблиц) будем применять правила из заранее сформулированного списка.

Доказательства такого вида это преобразование записей согласно заданному своду правил — принято называть *логическим выводом*.

Логический вывод, в котором преобразуются семантические таблицы, принято называть табличным выводом, и соответствующие правила преобразования — правилами табличного вывода.

Начнём с определения свода этих правил.

3.3. Правила табличного вывода

Будем использовать правила табличного вывода двух видов:

(*): $\dfrac{T_0}{T_1}$ (**): $\dfrac{T_0}{T_1, T_2}$

где T_0, T_1, T_2 — семантические таблицы.

Согласно правилу (*), рассматриваемая таблица T_0 преобразуется в таблицу T_1 для последующего рассмотрения, (**) в таблицы T_1 и T_2 для поочерёдного рассмотрения.

При этом правила будут подобраны так, чтобы в правилах (*) таблица T_0 была выполнима тогда и только тогда, когда и T_1 а в правилах (**) таблица T_0 была выполнима тогда и только тогда, когда выполнима хотя бы одна из таблиц T_1, T_2

Таблицы T_1, T_2 под чертой в правилах (**) иногда называют альтернативами.

Включим в свод 12 правил табличного вывода: 12=2·6, где 2 это части таблицы: левая, правая, а 6 число логических операций: $\&, \vee, \rightarrow, \neg, \forall, \exists$.
Согласно каждому правилу, в одной из частей таблицы выбирается одна формула, и эта формула преобразуется в одну или несколько в зависимости от её вида и расположения.
В правилах будут использоваться следующие обозначения:
Γ, Δ — произвольные множества формул;
ϕ, ψ — произвольные формулы;
x — произвольная предметная переменная;
t — произвольный терм, такой что подстановка $\{x/t\}$ правильна для ϕ;
c — произвольная константа, не содержащаяся в формулах из $\Gamma \cup \Delta \cup \{\phi\}$;

L&: $\langle \Gamma,\phi\&\psi \mid \Delta\rangle$ — чтобы формулу $\phi\&\psi$ сделать истинной необходимо сделать истинными обе формулы ϕ и ψ

$$\frac{\langle \Gamma,\phi\&\psi \mid \Delta\rangle}{\langle \Gamma,\phi,\psi \mid \Delta\rangle}$$

Lv: $\langle \Gamma, \phi \vee \psi \mid \Delta\rangle$ — чтобы формулу $\phi \vee \psi$ сделать истинной необходимо сделать истинной одну из формул ϕ или ψ

$$\frac{\langle \Gamma, \phi \vee \psi \mid \Delta\rangle}{\langle \Gamma,\phi \mid \Delta\rangle, \quad \langle \Gamma,\psi \mid \Delta\rangle}$$

L→: $\langle \Gamma,\phi\to\psi \mid \Delta\rangle$ — чтобы формулу $\phi\to\psi$ сделать истинной необходимо сделать истинной ψ или ложной ϕ

$$\frac{\langle \Gamma,\phi\to\psi \mid \Delta\rangle}{\langle \Gamma,\psi \mid \Delta\rangle,\langle \Gamma \mid \Delta,\phi\rangle}$$

L¬: $\langle \Gamma,\neg\phi \mid \Delta\rangle$ — чтобы формулу $\neg\phi$ сделать истинной необходимо сделать ложной формулу ϕ

$$\frac{\langle \Gamma,\neg\phi \mid \Delta\rangle}{\langle \Gamma \mid \Delta,\phi\rangle}$$

L∀: $\langle \Gamma,\forall x\ \phi \mid \Delta\rangle$ — чтобы формулу $\forall x\ \phi$ сделать истинной необходимо сделать истинными **все** формулы вида $\phi\{x/t\}$

$$\frac{\langle \Gamma,\forall x\ \phi \mid \Delta\rangle}{\langle \Gamma, \forall x\ \phi, \phi\{x/t\} \mid \Delta\rangle}$$

Здесь t - терм свободный для подстановки в формулу ϕ вместо переменной x.

L∃: $\langle \Gamma, \exists x\ \phi \mid \Delta\rangle$ — чтобы формулу $\exists x\ \phi$ сделать истинной необходимо сделать истинной формулу ϕ(**c**) с новой константой **c.**

$$\frac{\langle \Gamma, \exists x\ \phi \mid \Delta\rangle}{\langle \Gamma, \phi\{x/c\} \mid \Delta\rangle}$$

Rv: $\langle \Gamma \mid \Delta, \phi \vee \psi\rangle$ — чтобы формулу $\phi \vee \psi$ сделать ложной необходимо сделать ложной и ϕ и ψ

$$\frac{\langle \Gamma \mid \Delta, \phi \vee \psi\rangle}{\langle \Gamma \mid \Delta, \phi, \psi\rangle}$$

R→: $\langle \Gamma \mid \Delta,\phi\to\psi\rangle$ — чтобы формулу $\phi\to\psi$ сделать ложной необходимо сделать истинной формулу ϕ и ложной формулу ψ

$$\frac{\langle \Gamma \mid \Delta,\phi\to\psi\rangle}{\langle \Gamma,\phi \mid \Delta,\psi\rangle}$$

R¬: $\dfrac{\langle \Gamma \mid \Delta, \neg\phi \rangle}{\langle \Gamma, \phi \mid \Delta \rangle}$ чтобы формулу ¬φ сделать ложной необходимо сделать истинной формулу φ

R∀: $\dfrac{\langle \Gamma \mid \Delta, \forall x\ \phi \rangle}{\langle \Gamma \mid \Delta, \phi\{x/c\} \rangle}$ чтобы формулу ∀x φ сделать ложной необходимо сделать ложной формулу φ(**c**) с новой константой **c**

R∃: $\dfrac{\langle \Gamma \mid \Delta, \exists x\ \phi \rangle}{\langle \Gamma \mid \Delta, \exists x\ \phi, \phi\{x/t\} \rangle}$ чтобы формулу ∃x φ сделать ложной необходимо сделать ложными **все** формулы вида φ{x/t}

Здесь t - терм свободный для подстановки в формулу φ вместо переменной x.

Несколько слов об ограничениях на терм t и константу c в правилах L∀, R∀, L∃, R∃ .

1). Если разрешить подставлять любые термы в L∀, R∃, то можно столкнуться с потерей эквивалентности таблиц:
так $\langle \forall x\ \exists y\ P(x, y) \mid \exists y\ P(y, y) \rangle$ — выполнимая таблица, а после неправильного применения правила L∀
$\langle \forall x\ \exists y\ P(x, y), \exists y\ P(y, y) \mid \exists y\ P(y, y) \rangle$ — невыполнимая таблица.

2). Если разрешить подставлять «использованные» константы в L∃, R∀:
$\langle \exists x\ P(x) \mid P(c) \rangle$ — выполнимая таблица,
$\langle P(c) \mid P(c) \rangle$ — после «применения L∃» - невыполнимая таблица.

Т.о. если не соблюдать ограничения на термы и константы в указанных выше правилах, мы, используя эти правила, рискуем получать некорректные заключения о выполнимости таблиц.

3.4. Табличный вывод

Табличный вывод для таблицы T_0 — это размеченное корневое ориентированное дерево следующего вида, в котором все таблицы, приписанные листьям, закрыты или атомарны (в том числе могут быть одновременно закрытыми и атомарными):

$$
\begin{tikzcd}[column sep=small]
 & & T_0 \arrow[dl] \arrow[dr, "L\to"'] & & \\
 & T_i \arrow[dl] \arrow[dr, "L\vee"'] & & T_1 \arrow[dr, "R\to"'] & \\
T_j \arrow[d, "L\neg"'] & & T_k \arrow[d, "L\&"] & & T_2 \\
T_3 & & T_4 & & \text{закр.таб.} \\
\text{атом.таб.} & & \text{закр.атом.таб.} & &
\end{tikzcd}
$$

Табличный вывод успешен, если он конечен и всем его листьям приписаны закрытые таблицы.

Успешный табличный вывод явно демонстрирует, что таблица, для которой он построен, невыполнима. В частности, согласно теореме о табличной проверке общезначимости, если этот вывод построен для таблицы $\langle \mid \phi\rangle$, то верно $\models \phi$.

Приведём несколько примеров табличных выводов.

Пример 1.

$$
\begin{tikzcd}[column sep=small]
 & \langle \mid (\varphi\to\neg\psi)\to(\psi\to\neg\varphi)\rangle \arrow[d, "R\to"] & \\
 & \langle \varphi\to\neg\psi \mid \psi\to\neg\varphi\rangle \arrow[d, "R\to"] & \\
 & \langle \varphi\to\neg\psi, \psi \mid \neg\varphi\rangle \arrow[d, "R\neg"] & \\
 & \langle \varphi\to\neg\psi, \psi, \varphi \mid \rangle \arrow[dl] \arrow[dr, "L\to"'] & \\
\langle \neg\psi, \psi, \varphi \mid \rangle \arrow[d, "L\neg"] & & \langle \psi, \varphi \mid \varphi\rangle \\
\langle \psi, \varphi \mid \psi\rangle & & \text{закрытая таблица} \\
\text{закрытая таблица} & &
\end{tikzcd}
$$

Вывод успешен

При этом для любых формул φ, ψ верно $\models (\varphi\to\neg\psi)\to(\psi\to\neg\varphi)$

Пример 2.

$$\begin{array}{c}
\langle \mid \exists x\ P(x) \to \forall x\ P(x) \rangle \\
\downarrow R\to \\
\langle \exists x\ P(x) \mid \forall x\ P(x) \rangle \\
\downarrow L\exists \\
\langle P(\mathbf{c_1}) \mid \forall x\ P(x) \rangle \\
\downarrow R\forall \\
\langle P(\mathbf{c_1}) \mid P(\mathbf{c_2}) \rangle
\end{array}$$

Незакрытая атомарная таблица

Вывод неуспешен

При этом $\not\models \exists x\ P(x) \to \forall x\ P(x)$

Пример 3.

$$\begin{array}{c}
\langle \mid \forall x\ (A(x) \to B(x)) \to (\forall x\ A(x) \to \forall x\ B(x)) \rangle \\
\downarrow R\to \\
\langle \forall x\ (A(x) \to B(x)) \mid \forall x\ A(x) \to \forall x\ B(x) \rangle \\
\downarrow R\to \\
\langle \forall x\ (A(x) \to B(x)), \forall x\ A(x) \mid \forall x\ B(x) \rangle \\
\downarrow R\forall \\
\langle \forall x\ (A(x) \to B(x)), \forall x\ A(x) \mid B(\mathbf{c}) \rangle \\
\downarrow L\forall \\
\langle \forall x\ (A(x) \to B(x)), \forall x\ A(x), A(\mathbf{c}) \mid B(\mathbf{c}) \rangle \\
\downarrow L\forall \\
\langle \forall x\ (A(x) \to B(x)), \forall x\ A(x), A(\mathbf{c}) \to B(\mathbf{c}), A(\mathbf{c}) \mid B(\mathbf{c}) \rangle \\
\swarrow \quad L\to \quad \searrow
\end{array}$$

$$\left\langle \begin{array}{l} \forall x\ (A(x) \to B(x)), \\ \forall x\ A(x), B(\mathbf{c}), A(\mathbf{c}) \end{array} \mid B(\mathbf{c}) \right\rangle \qquad \left\langle \begin{array}{l} \forall x\ (A(x) \to B(x)), \\ \forall x\ A(x), A(\mathbf{c}) \end{array} \mid B(\mathbf{c}), A(\mathbf{c}) \right\rangle$$

Закрытая таблица — Закрытая таблица

Вывод успешен

При этом $\models \forall x\ (A(x) \to B(x)) \to (\forall x\ A(x) \to \forall x\ B(x))$

Пример 4.

$$\langle\ |\ \forall x\ \exists y\ P(x,y) \to \exists y\ \forall x\ P(x,y)\rangle$$
$$\downarrow R\to$$
$$\langle \forall x\ \exists y\ P(x,y)\ |\ \exists y\ \forall x\ P(x,y)\rangle$$
$$\downarrow L\forall$$
$$\langle \forall x\ \exists y\ P(x,y), \exists y\ P(\mathbf{c_1},y)\ |\ \exists y\ \forall x\ P(x,y)\rangle$$
$$\downarrow R\exists$$
$$\langle \forall x\ \exists y\ P(x,y), \exists y\ P(\mathbf{c_1},y)\ |\ \exists y\ \forall x\ P(x,y), \forall x\ P(x,\mathbf{c_2})\rangle$$
$$\downarrow L\exists$$
$$\langle \forall x\ \exists y\ P(x,y), P(\mathbf{c_1},\mathbf{c_3})\ |\ \exists y\ \forall x\ P(x,y), \forall x\ P(x,\mathbf{c_2})\rangle$$
$$\downarrow R\forall$$
$$\langle \forall x\ \exists y\ P(x,y), P(\mathbf{c_1},\mathbf{c_3})\ |\ \exists y\ \forall x\ P(x,y), P(\mathbf{c_4},\mathbf{c_2})\rangle$$
$$\downarrow L\forall$$
$$\infty$$

Вывод бесконечен (и, следовательно, неуспешен)

При этом $\not\models \forall x\ \exists y\ P(x,y) \to \exists y\ \forall x\ P(x,y)$

Пример 5.

$$\langle\ |\ \exists x\ \forall y\ P(x,y) \to \forall y\ \exists x\ P(x,y)\rangle$$
$$\downarrow R\to$$
$$\langle \exists x\ \forall y\ P(x,y)\ |\ \forall y\ \exists x\ P(x,y)\rangle$$
$$\downarrow L\exists$$
$$\langle \forall y\ P(\mathbf{c_1},y)\ |\ \forall y\ \exists x\ P(x,y)\rangle$$
$$\downarrow R\forall$$
$$\langle \forall y\ P(\mathbf{c_1},y)\ |\ \exists x\ P(x,\mathbf{c_2})\rangle$$
$$\downarrow L\forall$$
$$\langle \forall y\ P(\mathbf{c_1},y), P(\mathbf{c_1},\mathbf{c_3})\ |\ \exists x\ P(x,\mathbf{c_2})\rangle$$
$$\downarrow R\exists$$
$$\langle \forall y\ P(\mathbf{c_1},y), P(\mathbf{c_1},\mathbf{c_3})\ |\ \exists x\ P(x,\mathbf{c_2}), P(\mathbf{c_4},\mathbf{c_2})\rangle$$
$$\downarrow L\forall$$
$$\infty$$

Вывод бесконечен (и, следовательно, неуспешен)

При этом $\models \exists x\ \forall y\ P(x,y) \to \forall y\ \exists x\ P(x,y)$

Этот вывод легко переделывается в конечный, если при применении правила LA к четвертой таблице $< \forall y\ P(c1,y) > | \exists x\ P(x,c2) >$ в качестве терма t выбрать не

c3, а c2 и получить таблицу $<\ \forall y\ P(c1,y),\ \ P(c1,c2)\ |\ \exists x\ P(x,c2)\ >$, из которой замкнутая таблица получается за 1 шаг по правилу R∃, в котором в качестве тема берется c1.
Отсюда делаем вывод, что неправильный выбор термов в правилах LA и R∃ может сделать потенциально успешный вывод бесконечным применением различных правил вывода, не приводящим ни к какому результату. А как выбирать правильно термы в этих правилах так, чтобы для всех невыполнимых таблиц строился бы успешный вывод? К сожалению, в общем случае ответ на этот вопрос - «правила для выбора правильных термов не существует!». Т.о. процедуры, позволяющей для любой формулы логики предикатов определить, является ли она тавтологией, не существует», т.е. логика предикатов «не разрешима». Подробнее об этом ниже.

Пример 6. Проверить справедливость логического следствия

$\forall x\ \exists y\ A(y, x)\ \&\ \forall x\ \forall y\ \forall z(A(x, y)\ \&\ A(x, z) \to A(y, z)) \models$
$\forall x\ \forall y(A(x, y) \to A(y, x))$.

Обозначим $F1 = \forall x\ \exists y\ A(y, x)$

$F2 = \forall x\ \forall y\ \forall z(A(x, y)\ \&\ A(x, z) \to A(y, z))$

$G = \forall x\ \forall y(A(x, y) \to A(y, x))$.

Рассмотрим семантическую таблицу

T0: $<F1,\ F2\ |\ G>$ и построим для неё табличный вывод

T1: $<F1, F2\ |\ A(a,b) \to A(b,a)>$ - получена по правилу R∀, примененному 2 раза x=a, y=b

T2: $<F1, F2, A(a,b)\ |\ A(b,a)>$ - получена по правилу R→

T3: $<F1,F2, A(a,b),\ \exists y A(y,a)\ |\ A(b,a)>$ - получена по правилу L∀ с формулой F1

T4: $<F1,F2, A(a,b),\ A(c,a)\ |\ A(b,a)>$ - получена по правилу L∃

T5: $<F1,F2, A(a,b),\ A(c,a), A(a,b)\&A(a,a) \to A(b,a)\ |\ A(b,a)>$ - получена по правилу L∀ с формулой F2 3 раза x=a, y=b, z=a

T6*: $<F1,F2, A(a,b),\ A(c,a), A(b,a)\ |\ A(b,a)>$ - * -замкнутая, получена из T5 по правилу L→

T7: $<F1, F2, A(a,b), A(c,a)\ |\ A(b,a), A(a,b)\&A(a,a)>$ - получена из T5 по правилу L→

T8*: $<F1,\ F2, A(a,b),\ A(c,a)\ |\ A(b,a), A(a,b)>$ * замкнутая- получена из T7 по правилу R&

T9: $<F1, F2, A(a,b),\ A(c,a)\ |\ A(b,a), A(a,a)>$ - получена из T7 по правилу R&

T10: $<F1, F2, A(a,b),\ A(c,a), A(c,a)\&A(c,a) \to A(a,a)\ |\ A(b,a), A(a,a)>$ - получена по правилу L∀ из T9 с формулой F2 3 раза x=c, y=a, z=a

T11*: $< F1, F2, A(a,b), A(c,a), A(a,a) \mid A(b,a), A(a,a) >$ -замкнутая, получена из T10 по правилу L→

T12*: $< F1, F2, A(a,b), A(c,a) \mid A(b,a), A(a,a), A(c,a) >$- замкнутая, получена из T10 по правилу L→.

Построен успешный табличный вывод. Следствие справедливо.

Это пример удачного выбора термов в двух применениях правила L∀ к таблицам T5 и T10 соответственно.

3.5. Лемма (о корректности правил табличного вывода)

Для любого правила табличного вывода вида $\dfrac{T_0}{T_1(, T_2)}$,

а именно L&, R&, L∨, R∨, L→, R→, L¬, R¬, L∀, R∀, L∃, R∃ — таблица T_0 выполнима тогда и только тогда, когда выполнима таблица T_1 (или выполнима таблица T_2)

3.6. Теорема (о корректности табличного вывода)

Если для семантической таблицы T существует успешный табличный вывод, то таблица T невыполнима.

Следствие.

Если для таблицы $\langle \mid \phi \rangle$ существует успешный табличный вывод, то $\models \phi$.

3.7.Теорема (о полноте табличного вывода)

Для любой невыполнимой семантической таблицы существует успешный табличный вывод.

Следствие 1.

Семантическая таблица T логики предикатов невыполнима ⇔ для неё существует успешный табличный вывод.

Следствие 2. Для любой формулы ϕ логики предикатов верно: $\models \phi$ ⇔ для семантической таблицы $\langle \mid \phi \rangle$существует успешный табличный вывод.

3.8.Автоматизация построения табличного вывода

В свете того, что имеется стратегия построения успешного табличного вывода для произвольной невыполнимой таблицы, используемая в доказательстве теоремы о полноте, возникает вопрос: можно ли поручить проверку общезначимости формул ЛП компьютеру, чтобы он делал всю работу за нас?

Если формула общезначима, то это можно обосновать, придерживаясь упомянутой стратегии. А если не общезначима, то на этот счёт пока есть

только теорема о корректности: успешного вывода для соответствующей таблицы не существует.
Познакомившись получше с логикой предикатов, вы узнаете, что целиком переложить такую работу на компьютер невозможно.
Дело в том, что проблема установления общезначимости формул ЛП алгоритмически неразрешима. Т.е. не существует алгоритма, который за конечное число шагов по произвольной формуле логики предикатов мог бы определить, общезначима ли она или нет.

Но попытки найти эффективные методы установления общезначимости формул логики предикатов не прекращаются. Программы реализующие эти методы называются пруверами.

Представим себе прувер, способный проверять общезначимость формул логики предикатов методом семантических таблиц, работающий корректно и выдающий ответ «да» для всех общезначимых формул.
Насколько эффективен может быть такой прувер?
Эффективность построения табличного вывода определяется следующем:
1) как на каждом шаге выбираются формулы для применения к ним правил и
2) какие термы подставляются при применении правил L∀ и R∃.
Если при ответе на первый вопрос можно дать некоторые рекомендации, например, такие:
-в первую очередь применять правила, которые упрощают текущую таблицу и не приводят к необходимости построения альтернативных таблиц. Это правила L&, L¬, L∃, R¬, R→, R∨, R∀;
- во вторую очередь поменять правила, упрощающие таблицы, но требующие построения альтернативных таблиц. Это правила L∨, L→, R&.
То при применении правил L∀, R∃ необходимо стараться избегать перебора слишком большого числа термов, иначе такой прувер окажется очень неэффективным.
Избежать перебора большого числа термов — непростая задача.
Для примера представим себе, что при построении вывода
придётся перебрать все термы, составленные из одного функционального символа $f^{(2)}$, используемого не более 10 раз, и двух различных констант (вроде бы это не очень большие термы?) Можно легко посчитать, что существует более 10^{300} различных термов такого вида.
Число 10^{100} (на 200 нолей меньше) имеет особое название — **гугол**: это бессмысленно большое число, превосходящее число атомов в наблюдаемой части вселенной.

Существуют способы повышения эффективности перебора термов при построении логического вывода для доказательства общезначимости формул.
Далее обсудим один из таких способов и основанный на нём метод - метод резолюций.

3.9. Упражнения для самопроверки

Упражнение 1. Доказать общезначимость указанных ниже формул, построив успешные выводы соответствующих семантических таблиц:

1. $\forall x\, P(x) \rightarrow \forall y\, P(y)$;
2. $\neg\exists x P(x) \rightarrow \forall x \neg P(x)$;
3. $\forall x \neg P(x) \rightarrow \neg\exists x P(x)$;
4. $\forall x\, (P(x) \& R(x)) \rightarrow (\forall x\, P(x)\ \&\ \forall x\, R(x))$;
5. $(\forall x\, P(x)\ \&\ \forall x\, R(x)) \rightarrow \forall x\, (P(x) \& R(x))$;
6. $\exists x\, (P(x) \vee R(x)) \rightarrow (\exists x\, P(x) \vee \exists x\, R(x))$;
7. $(\exists x\, P(x) \vee \exists x\, R(x)) \rightarrow \exists x\, (P(x) \vee R(x))$;
8. $(\forall x\, P(x) \vee R(y)) \rightarrow \forall x\, (P(x) \vee R(y))$;
9. $\forall x\, (P(x) \vee R(y)) \rightarrow (\forall x\, P(x) \vee R(y))$;
10. $\exists y\, \forall x\, Q(x,y) \rightarrow \forall x\, \exists y\, Q(x,y)$;
11. $\forall x((\exists x \neg P(x) \rightarrow \exists x R(x)) \rightarrow \exists y(P(x) \vee R(y)))$;
12. $\forall x\, (P(x) \rightarrow \exists y\, R(x,f(y))) \rightarrow (\exists x\, \neg P(x) \vee \forall x \exists z R(x,z))$;
13. $\forall x\, \exists y\, \forall z\, (R(x, y) \rightarrow R(y, z))$;
14. $\exists x\, \forall y\, \exists z\, (R(x, y) \rightarrow R(y, z))$;
15. $\exists x\, (R(x)\ \&\ \exists x\, (P(x) \rightarrow \neg R(x)) \rightarrow \neg\forall x\, P(x))$;
16. $\exists x\, ((\forall y\, P(x,y) \vee \exists x\, R(x)) \rightarrow (\exists x\, P(x,x) \vee R(x)))$;
17. $\exists x\, (\forall x\, P(x) \rightarrow \neg(R(x)\ \&\ \exists x\, (P(x) \rightarrow \neg R(x))))$;
18. $\exists x\, (\exists y\, \neg P(x,y) \rightarrow \forall x\, R(x)) \rightarrow \forall x\, (R(x) \vee \exists x\, P(x,f(x)))$;
19. $\forall x \exists u(\exists v \forall y\, ((E(u, y) \rightarrow H(y, v))\ \&\ \exists w \forall x\, (H(w, y) \rightarrow \neg H(x, v))) \rightarrow \exists y\, \neg E(x, y))$;
20. $\forall x\, (\forall y \exists v \forall u\, ((A(u, v) \rightarrow B(y, u))\ \&\ (\neg\exists w\, A(w, u) \rightarrow \forall w\, A(w, v))) \rightarrow \exists y\, B(x, y))$.

Упражнение 2.

Выясните, применяя табличный вывод, какие из приведенных ниже формул являются общезначимыми, какие являются выполнимыми и какие - невыполнимыми:

1. $\neg(\exists x\, P(x) \rightarrow \forall x\, P(x))$;
2. $\exists x\, \forall y\, (Q(x, x) \& \neg Q(x, y))$;
3. $\exists x\, (P(x)\ \&\ \exists x\, \neg P(x))$;
4. $\forall x\, (P(x)\ \&\ \forall x\, \neg P(x))$;

5. $(\exists x\ P(x)\ \&\ \exists x\ R(x)) \rightarrow \exists x\ (P(x)\&R(x))$;
6. $(\forall x\ P(x)\ \&\ \forall x\ R(x)) \rightarrow \forall x\ (P(x)\&R(x))$.

Упражнение3. Какие из приведенных ниже множеств формул являются непротиворечивыми. Используйте выводимость соответствующих семантических таблиц.

1. $\Gamma_1 = \{\ \forall x\ \neg R(x,x), \exists x\ P(x), \forall x \exists y\ R(x,y), \forall x\ (P(x) \rightarrow R(y,x))\ \}$;
2. $\Gamma_2 = \{\ \forall x\ \neg R(x, x), \forall y \exists x\ R(y, x), \forall x \forall y \forall z\ (R(x, y)\&R(y, z) \rightarrow R(x, z))\ \}$.

3.10. Логика ALC

Приведем пример использования табличного вывода в построении онтологий - важного инструмента хранения и обработки знаний в важнейших прикладных областях, таких как медицина, юриспруденция и др. Развитые онтологии основываются на дескриптивных логиках, определяющих интеллектуальные возможности соответствующих онтологий. Одной из простых и распространенных дескриптивных логик является логика ALC.

Синтаксис ALC

Язык описания ALC**-концептов (классов)**

- имена концептов $A_0, A_1, \ldots$
- имена ролей $r_0, r_1, \ldots$
- концепт $\top$ (часто называют “вещь”)
- концепт $\bot$ (пустой класс)
- логическая связка $\sqcap$ (пересечение, конъюнкция, “и”).
- квантор $\exists$ (существование).
- квантор $\forall$ (часто называют ограничение значения (value restriction)).
- логическая связка $\sqcup$ (объединение, дизъюнкция,“или”).
- логическая связка $\neg$ (дополнение, отрицание).

ALC**-концепты** определяются индуктивно следующим образом:
Все имена концептов, $\top$ и $\bot$ являются ALC-концептами;
Если C является ALC-концептом, то и $\neg C$ является ALC-концептом.
Если C и D являются ALC-концептами,то $(C \sqcap D)$, $(C \sqcup D)$, являются ALC-концептами.
Если C является ALC-концептом, а r — имя роли, то $\exists r.C$, $\forall r.C$ являются ALC-концептами
ALC импликация концептов имеет вид $C \sqsubseteq D$, где C, D являются ALC-концептами.

ALC-**TBox** есть конечное множество импликаций ALC-концептов.

Семантика ALC

Интерпретацией называется структура $I = (\Delta^I, \cdot^I)$ в которой – Δ^I **носитель** (непустое множество), а $\cdot^I$ **функция**, отображающая
имя концепта A в подмножество $A^I \subseteq \Delta^I$;
имя роли r в бинарное отношение r^I на Δ^I, $(r^I \subseteq \Delta^I \times \Delta^I)$.
• Значение сложных концептов в интерпретации I:
– $(\top)^I=\Delta^I$ и $(\bot)^I=\varnothing$
– $(\neg C)^I=\Delta^I \backslash C^I$
– $(C \sqcap D)^I = C^I \cap D^I$ и $(C \sqcup D)^I = C^I \cup D^I$
– $(\forall r.C)^I = \{x \in \Delta^I$ | для всех $y \in \Delta^I$ $((x,y) \in r^I \to y \in C^I)\}$
– $(\exists r.C)^I = \{x \in \Delta^I$ | существует $y \in \Delta^I$ такой что $(x,y) \in r^I$ и $y \in C^I\}$
Пусть I — интерпретация, $C \sqsubseteq D$ — импликация ALC-концептов и T - ALC-TBox, $C \equiv D$ - сокращение для конъюнкции двух импликаций $C \sqsubseteq D$ и $D \sqsubseteq C$. Тогда

- $I \models C \sqsubseteq D$ т. и т.т., когда $C^I \subseteq D^I$.
- $I \models C \equiv D$ т. и т.т., когда $C^I \subseteq D^I$ и $D^I \subseteq C^I$.
- $I \models T$ т.ит.т., когда $I \models E \sqsubseteq F$ для всех $E \sqsubseteq F$ в T.

Пример.
Пусть задан T-box :
Woman, Person, Female, Mother, Parent - имена концептов,
hasChild - имя роли.

Набор имликаций:
1. Woman ≡ Person ⊓ Female ,
2. Mother ≡ Parent ⊓ Female
3. Parent ≡ Person ⊓ ∃hasChild.Person

Спрашивается, поглощает ли концепт Women концепт Mother, то есть справедлива ли импликация **Mother ⊑ Women** ?

Подобные задачи естественным образом сводятся к задаче построения табличного вывода для семантических таблиц.

Так для каждого имени концепта Woman, Person, Female, Mother, Parent введем одноместный предикатный символ W(x), Per(x), F(x), M(x), Par(x) соответственно. Для каждого имени роли hasChild (здесь оно одно) введем двухместный предикатный символ HC(x,y).

Запишем набор импликаций из Tbox'а в виде формул логики предикатов:
F1: $\forall x\, (W(x) \equiv Per(x) \;\&\; F(x))$

F2: $\forall x\ (M(x) \equiv Par(x)\ \&\ F(x))$

F3: $\forall x\ (Par(x) \equiv \exists y\ (HC\ (x,y)\ \&\ Per\ (y))$

Задаче о поглощении концептом Woman концепта соответствует формула:

G: $\forall x\ (\ M(x) \to W(x))$.

Решением задачи будет проверка истинности логического следствия

F1, F2, F3 $\vDash$ G или проверка невыполнимости семантической таблицы

$<$F1, F2, F3 | G$>$, которая в данном случае будет иметь вид:

$< \forall x\ (W(x) \equiv Per(x)\ \&\ F(x)), \forall x\ (M(x) \equiv Par(x)\ \&\ F(x)),$

$\forall x\ (Par(x) \equiv \exists y\ (HC\ (x,y)\ \&\ Per\ (y))\ |\ \forall x\ (\ M(x) \to W(x)) >$.

Можно проверить, что для данной семантической таблицы успешный вывод строится. Достаточно использовать естественную стратегию поиска: сначала применяются правила для логических связок, затем правила L$\exists$, R$\forall$, вводящие новые константы, и лишь затем правила L$\forall$, R$\exists$ с термами равными уже введенным константам.

Заметим, что эта стратегия является полной для всех семантических таблиц, возникающих при анализе ALC-Tbox'ов. Разработаны модификации этой стратегии для анализа более выразительных дескриптивных логик.

Подчеркнем еще раз, что на основе дескриптивных логик (являющихся, как правило, расширением логики ALC) разрабатываются популярные в настоящее время системы представления знаний о предметных новостях, называемые *онтологиями.* Интеллектуальность подобных систем основана на их способности отвечать на продвинутые вопросы в данной предметной области. Это обеспечивается специальными программами, называемые *пруверами* или *ризонерами*, которые, по-существу, ищут обоснование импликаций концептов в Tbox-е, описывающих соответствующую предметную область, т.е.уу у вывод соответствующих семантических таблиц.

Глава 4. Основные эквивалентности (равносильности) формул логики предикатов. Предвоенная нормальная форма. Сколемовская стандартная форма. Системы дизъюнктов.

В логике высказываний, пользуясь основными эквивалентностями, мы выполняли преобразования произвольных форму вида:

$A \vee B \to (C \,\&\, D \to E)$ ~ $(x \to y \sim \neg x \vee y)$

$\neg(A \vee B) \vee \neg(C \,\&\, D) \vee E$ ~ $(\neg(x \,\&\, y) \sim \neg x \vee \neg y)$

$(A \,\&\, \neg B) \vee \neg C \vee \neg D \vee E$ ~

$(A \vee \neg C \vee \neg D \vee E) \,\&\, (\neg B \vee \neg C \vee \neg D \vee E)$

Нижняя формула получена из верхней при помощи основных тождеств логики высказываний. Значит, можно быть уверенным в том,
что эти формулы имеют одинаковый смысл
Неплохо было бы уметь преобразовывать формулы логики предикатов с гарантированным сохранением их смысла.

4.1. Равносильность формул логики предикатов

Эквивалентность (логическая связка):
$\phi \equiv \psi$ — это сокращение для формулы $(\phi \to \psi) \& (\psi \to \phi)$
Будем говорить, что формулы ϕ и ψ равносильны ($\phi \sim \psi$), если формула
$\phi \equiv \psi$ общезначима.

Утверждение. Для любых равносильных формул $\phi(\tilde{x}^n)$, $\psi(\tilde{x}^n)$ логики предикатов, интерпретации I и набора предметов $\tilde{d}^n$ верно следующее:

$I \models \phi[\tilde{d}^n] \quad \Leftrightarrow \quad I \models \psi[\tilde{d}^n]$

Доказательство.

($\Rightarrow$) Пусть $I \models \phi[\tilde{d}^n]$.

Тогда: $\models (\phi \to \psi) \& (\psi \to \phi)$ (по определению равносильности)

$I \models ((\phi \to \psi) \& (\psi \to \phi))[\tilde{d}^n]$ (по определению общезначимости)

$I \models (\phi \to \psi)[\tilde{d}^n]$ (по семантике «&»)

$I \models \psi[\tilde{d}^n]$ (по семантике «$\to$» и соотношению $I \models \phi[\tilde{d}^n]$)

($\Leftarrow$) Аналогично

Утверждение. Если формула ϕ общезначима, то любая равносильная ей формула ψ также общезначима.

Утверждение. Если формула ϕ выполнима, то любая равносильная ей формула ψ также выполнима

4.2 Основные равносильности ЛП

Они включают в себя основные эквивалентности логики высказываний и правила работы с кванторами:

1. Переименование связанной переменной:

$\forall x\ \phi \sim \forall y\ (\phi\{x/y\})$ $\exists x\ \phi \sim \exists y\ (\phi\{x/y\})$

(если $y \notin Var_{\phi}$ и подстановка $\{x/y\}$ правильна для ϕ)

2. Продвижение отрицания под квантор:

$\neg\forall x\ \phi \sim \exists x\ \neg\phi$ $\neg\exists x\ \phi \sim \forall x\ \neg\phi$

3.Вынесение квантора за скобки:

$\forall x\ \phi\ \&\ \psi \sim \forall x\ (\phi\ \&\ \psi)$ $\exists x\ \phi\ \&\ \psi \sim \exists x\ (\phi\ \&\ \psi)$

$\forall x\ \phi \vee \psi \sim \forall x\ (\phi \vee \psi)$ $\exists x\ \phi \vee \psi \sim \exists x\ (\phi \vee \psi)$

(если $x \notin Var_{\psi}$)

Для обоснования этих равносильностей достаточно применить метод семантических таблиц.

Введем следующие обозначения

$\phi[\![\psi]\!]$ — обозначение формулы ϕ, содержащей подформулу ψ ,

$\phi[\![\psi/\chi]\!]$ — формула, получающаяся из ϕ заменой некоторого вхождения подформулы ψ на формулу χ.

4.3. Теорема о равносильной замене

Теорема (о равносильной замене в ЛП).

Для любых формул ϕ, ψ, χ логики предикатов верно:

$\psi \sim \chi \Rightarrow \phi[\![\psi]\!] \sim \phi[\![\psi/\chi]\!]$

Пример.

Используя равносильные замены, можно существенно изменить форму высказывания, сохранив его смысл — (не)выполнимость в каждой интерпретации на каждом наборе предметов.

$\forall x\ P(x) \rightarrow \exists x\ P(x) \sim$

$(\exists x\ \phi \sim \exists y\ (\phi\{x/y\}))$

$\forall x\ P(x) \rightarrow \exists y\ P(y) \sim$

$(\phi \rightarrow \psi \sim \neg\phi \vee \psi)$

$\neg\forall x\ P(x) \vee \exists y\ P(y) \sim$

$(\neg\forall x\ \phi \sim \exists x\ \neg\phi)$

$\exists x\ \neg P(x) \vee \exists y\ P(y) \sim$

$(\exists x\ \phi \vee \psi \sim \exists x\ (\phi \vee \psi)$

$\exists x\ \exists y\ (\neg P(x) \vee P(y)) \sim$

$(\phi \rightarrow \psi \sim \neg\phi \vee \psi)$

$\exists x \exists y\ (P(x) \rightarrow P(y))$

4.4. Предварённая нормальная форма

Замкнутая формула логики предикатов находится в предварённой нормальной форме (ПНФ), если она имеет вид

$Q_1x_1 \dots Q_nx_n (D_1 \& \dots \& D_k)$, где

$Q_1x_1 \dots Q_nx_n$ кванторная приставка, $Q_1,\dots,Q_n \in \{\forall,\exists\}$

$(D_1 \& \dots \& D_k)$ матрица — это формула без кванторов в конъюнктивной нормальной форме (КНФ):

-$D_i = L^i_1 \vee \dots \vee L^i_{m_i}$ —множитель

-L^i_j — литера: атом или его отрицание

Наряду с «находится в ПНФ» будем говорить «является ПНФ»

Пример.

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

находится в предварённой нормальной форме:

$\forall x\, \exists y\, \exists z\, \forall u$ - кванторная приставка,

$P(x) \,\&\, \neg R(x, u) \,\&(\neg P(y) \vee R(x, z))$ - матрица;

Множители: $P(x)$, $\neg R(x,u)$, $\neg P(y) \vee R(x,z)$;

Литеры: $P(x)$, $\neg R(x,u)$,$\neg P(y)$, $R(x,z)$.

Теорема о предварённой нормальной форме.

Для любой замкнутой формулы логики предикатов существует равносильная ей предварённая нормальная форма.

Доказательство.

Опишем способ приведения произвольной формулы к ПНФ при помощи применения основных равносильностей логики предикатов.

Шаги приведения сгруппируем в 5 этапов.

Проиллюстрируем устройство этапов на примере конкретной формулы:

$\neg\exists x\, (P(x) \,\&(\forall x\, P(x) \to \exists y\, R(x, y)) \to \exists y\, R(x, y))$

1. Переименование переменных.

Переименуем связанные переменные так,

чтобы разными кванторами связывались разные переменные.

Для этого применим основные равносильности

$\forall x\, \phi \sim \forall y\, (\phi\{x/y\})$ и $\exists x\, \phi \sim \exists y\, (\phi\{x/y\})$,

каждый раз выбирая «новую» переменную y

$\neg\exists x\, (P(x) \,\&(\forall x\, P(x) \to \exists y\, R(x, y)) \to \exists y\, R(x, y)) \sim$

$\neg\exists x\, (P(x) \,\&(\forall z\, P(z) \to \exists y\, R(x, y)) \to \exists y\, R(x, y)) \sim$

$\neg\exists x\, (P(x) \,\&(\forall z\, P(z) \to \exists y\, R(x, y)) \to \exists u\, R(x, u))$

2. Удаление импликаций.

Удалим из формулы все импликации при помощи основной равносильности $\phi \to \psi \sim \neg\phi \vee \psi$:

$\neg\exists x\ (P(x)\ \&(\forall z\ P(z) \to \exists y\ R(x, y)) \to \exists u\ R(x, u)) \sim$

$\neg\exists x\ (P(x)\ \&(\neg\forall z\ P(z) \vee \exists y\ R(x, y)) \to \exists u\ R(x, u)) \sim$

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

3. Продвижение отрицаний.

Преобразуем формулу так,
чтобы отрицания располагались только непосредственно над атомами
Для этого применим основные равносильности

$\neg(\phi\ \&\ \psi) \sim \neg\phi \vee \neg\psi \quad \neg(\phi \vee \psi) \sim \neg\phi\ \&\ \neg\psi \quad \neg\neg\phi \sim \phi$

$\neg\forall x\ \phi \sim \exists x\ \neg\phi \qquad \neg\exists x\ \phi \sim \forall x\ \neg\phi$

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

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

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

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

4. Вынесение кванторов.

Вынесем все кванторы «наружу», собрав их в кванторную приставку.
Для этого применим основные равносильности

$\forall x\phi\&\psi\sim\forall x(\phi\&\psi) \qquad \exists x\phi\&\psi\sim\exists x(\phi\&\psi). \quad \chi1\&\chi2 \sim\chi2\&\chi1$

$\forall x\phi\vee\psi\sim\forall x(\phi\vee\psi) \qquad \exists x\phi\vee\psi\sim\exists x(\phi\vee\psi) \qquad \chi1 \vee\chi2 \sim\chi2 \vee\chi1$

После этапов 2, 3 «над» кванторами могут располагаться только & и $\vee$. После этапа 1 при вынесении квантора за скобки в ψ не встречается переменная этого квантора.

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

$\sim\ \forall x\ (P(x)\ \&\ \_\exists z\ (\neg P(z) \vee \exists y\ R(x, y))\ \&\ \forall u\ \neg R(x, u))$

$\sim\ \forall x\ (\exists z\ (P(x)\ \&\ \_(\neg P(z) \vee \exists y\ R(x, y)))\ \&\ \forall u\ \neg R(x, u))$

$\sim \ldots \sim\ \forall x\ \exists z\ \exists y\ \forall u\ (P(x)\ \&(\neg P(z) \vee R(x, y))\ \&\ \neg R(x, u))$

5. Получение КНФ.

С использованием законов булевой алгебры

$\psi \vee (\chi_1\ \&\ \chi_2) \sim (\psi \vee \chi_1)\ \&(\psi \vee \chi_2) \quad \psi \vee \chi \sim \chi \vee \psi$

матрицу формулы можно легко преобразовать в КНФ.

В рассматриваемом примере никакие преобразования не нужны, а методы приведения произвольной булевой формулы к КНФ должны быть вам уже известны.

Итог: после этапа 4 в формуле появляется кванторная приставка, а после этапа 5 «под» приставкой располагается КНФ,
то есть получается ПНФ.

4.5. Сколемовская стандартная форма. Лемма об удалении квантора существования. Алгоритм сколемизации

Замкнутая формула логики предикатов находится в сколемовской стандартной форме (ССФ), если

1) она находится в предварённой нормальной форме;

2) её кванторная приставка не содержит кванторов $\exists$ и имеет вид

$\forall \tilde{x}^n (D_1 \& \ldots \& D_k)$.

Пример. Формула

$\forall x\ \forall u\ (P(x) \& (\neg P(f(x)) \vee R(x, g(x))) \& \neg R(x, u))$

находится в сколемовской стандартной форме.

Наряду с «находится в ССФ» будем говорить «является ССФ».

<u>Лемма об удалении квантора существования.</u>

Пусть $\phi = \forall \tilde{x}^n\ \exists x_{n+1}\ \chi$ — замкнутая формула логики предикатов ($n \geq 0$) и функциональный символ f не содержится в χ. Тогда

$\models \phi \Leftrightarrow\ \models \forall \tilde{x}^n\ (\chi\{x_{n+1}/f(\tilde{x}^n)\})$.

Небольшая вольность: если слева от $\exists$ не стоит ни одного $\forall$, то, согласно лемме, f — 0-местный функциональный символ: так будем называть константы, и писать «f()» наряду с «f».

Устранение квантора $\exists$ с введением новых функциональных символов с целью получить более простую «хорошую» формулу обычно называют сколемизацией (здесь «хорошая» — сохраняющая выполнимость и невыполнимость).

При устранении $\exists$ на место удаляемой переменной подставляются сколемовские термы (здесь — $f(\tilde{x}^n)$)

<u>Алгоритм сколемизации ПНФ.</u>

Дано: ПНФ ϕ_{pnf}

Требуется получить ССФ $Sk(\phi_{pnf})$, такую что

ϕ_{pnf} выполнима $\Leftrightarrow Sk(\phi_{pnf})$ выполнима

<u>Алгоритм.</u>

Пока в кванторной приставке есть хотя бы один квантор $\exists$, самый левый $\exists$ удаляется при помощи подстановки сколемовского терма:

$\phi pnf = \forall x1 \ldots \forall xk{-}1\ \exists xk\ \forall xk{+}1 \ldots \forall xm{-}1\ \exists xm \ldots \chi \mapsto$
$\forall x1 \ldots \forall xk{-}1\ \forall xk{+}1 \ldots \forall xm{-}1\ \exists xm \ldots (\chi\{xk/\mathbf{f1(x1,\ldots,xk{-}1)}\}) \mapsto$
$\forall x1 \ldots \forall xk{-}1\ \forall xk{+}1 \ldots \forall xm{-}1 \ldots$
$(\chi\{xk/\mathbf{f1(x1,\ldots,xk{-}1)}, xm/\mathbf{f2(x1,\ldots,xk{-}1, xk{+}1,\ldots,xm{-}1)}\}) \mapsto$
$\ldots Sk(\phi pnf)$.

При этом важно при удалении очередного $\exists$ каждый раз выбирать новый функциональный символ, не содержащийся в формуле после всех предыдущих удалений

Пример 1:

$\forall x\ \exists z\ \exists y\ \forall u\ (P(x) \,\&(\neg P(z) \vee R(x,y)) \,\&\, \neg R(x,u)) \mapsto$
$\forall x\ \exists y\ \forall u\ (P(x) \,\&(\neg P(\mathbf{f}\,(x)) \vee R(x,y)) \,\&\, \neg R(x,u)) \mapsto$
$\forall x\ \forall u\ (P(x) \,\&(\neg P(\mathbf{f}\,(x)) \vee R(x,\mathbf{g}(x))) \,\&\, \neg R(x,u))$.

Ограничения на выбор символов в сколемовских термах:
f — любой функциональный символ местности 1 ,
g — любой функциональный символ местности 1, кроме f

Пример 2.

$\exists u\ \exists v\ \forall w\ \exists x\ \forall y\ \exists z\ (P(u,v)\&P(v,w)\&P(w,x)\&P(x,y)\&P(y,z)\&P(z,u)) \mapsto$
$\exists v\ \forall w\ \exists x\ \forall y\ \exists z\ (P(c,v)\&P(v,w)\&P(w,x)\&P(x,y)\&P(y,z)\&P(z,c)) \mapsto$
$\forall w\ \exists x\ \forall y\ \exists z\ (P(c,d)\&P(d,w)\&P(w,x)\&P(x,y)\&P(y,z)\&P(z,c)) \mapsto$
$\forall w\ \forall y\ \exists z\ (P(c,d)\&P(d,w)\&P(w,f(w))\&P(f(w),y)\&P(y,z)\&P(z,c)) \mapsto$
$\forall w\ \forall y\ (\ P(c,d)\&P(d,w)\&P(w, f(w))\&P(f\,(w), y) \,\&\, P(y, g(w, y)) \,\&$
$\quad\quad \& \, P(g(w, y), c)\ )$.

Ограничения на выбор символов в сколемовских термах:
c — любая константа (ни одной не содержится в формуле);
d — любая константа, кроме c;
f — любой функциональный символ местности 1 (ни один не содержится в формуле);
g — любой функциональный символ местности 2 (ни один не содержится в формуле).

Теорема (о сколемизации).
Для любой ПНФ ϕ_{pnf} формула $Sk(\phi_{pnf})$ является ССФ, для которой верно следующее: $\models \phi_{pnf} \Leftrightarrow\ \models Sk(\phi_{pnf})$.

Доказательство.
Пусть $\psi_1, \psi_2, \ldots, \psi_k$ — формулы, последовательно получающиеся удалением кванторов $\exists$ (по одному) согласно алгоритму сколемизации ($\psi_1 = \phi_{pnf}$, $\psi_k = Sk(\phi_{pnf})$)
По лемме об удалении квантора $\exists$, справедливы равносильности
$\models\phi_{pnf} \Leftrightarrow \models\psi_2 \Leftrightarrow \ldots \Leftrightarrow \models Sk(\phi_{pnf})$.

4.6. Системы дизъюнктов

Дизъюнктом называется ССФ с одним множителем в матрице: $\forall\tilde{x}^n (L_1 \vee\cdots\vee L_k)$, где L_i — литера (атом или его отрицание).
Для краткости иногда будем опускать кванторную приставку дизъюнктов: $\forall\tilde{x}^n (L_1 \vee\cdots\vee L_k) \mapsto L_1 \vee\cdots\vee L_k$
Для упрощения технических выкладок будем отождествлять между собой дизъюнкты, получающиеся друг из друга перестановкой слагаемых.
В связи с таким упрощением будем
отождествлять дизъюнкт с мультимножеством его литер:
$L_1\vee\cdots\vee L_k = \{L_1,\ldots,L_k\}$.
Пустой дизъюнкт $\square$- это особый дизъюнкт, представляющий собой пустое множество литер.
Пустой дизъюнкт будем считать невыполнимым.
Системой дизъюнктов будем называть (любое) множество дизъюнктов.
Утверждение. $\forall x\, (\phi\&\psi) \sim \forall x\, \phi\&\forall x\, \psi$.
Доказательство. Обосновать эту равносильность настолько же просто, как и все основные равносильности.
Теорема (о переходе к дизъюнктам)
Для ССФ с любым набором множителей $D_1,\ldots,D_k$ верно:

$\models \forall\tilde{x}^n (D_1 \&\ldots\&D_k) \Leftrightarrow \models \{\forall\tilde{x}^n D_1,\ldots,\forall\tilde{x}^n D_k\}$.

Доказательство.

По утверждению выше, $\forall\tilde{x}^n (D_1 \&\ldots\&D_k) \sim \forall\tilde{x}^n D_1 \&\ldots\&\forall\tilde{x}^n D_k$

Следовательно, с учётом семантики &,

$\models\forall\tilde{x}^n (D_1\&\ldots\&D_k) \Leftrightarrow$

$\models \forall\tilde{x}^n D_1 \&\ldots\&\forall\tilde{x}^n D_k \Leftrightarrow$

$\models \{\forall\tilde{x}^n D_1,\ldots,\forall\tilde{x}^n D_k\}$

Пример.

$\nVdash \forall x\ \forall u\ (P(x)\ \&(\neg P(f\ (x)) \vee R(x, g(x)))\ \&\ \neg R(x, u)) \Leftrightarrow$
$\nVdash \{P(x), \neg P(f\ (x)) \vee R(x, g(x)), \neg R(x, u)\}$

4.7. Упражнения для самопроверки

Упражнение 1. Используя основные равносильности, привести ниже указанные формулы к ПНФ:

$\exists x \forall y\ P(x,y)\ \&\ \forall x \exists y\ P(y,x)$;
$\forall x\ ((\exists y\ P(y,x) \to \exists y\ P(x,y)) \to Q(x)) \to \exists x\ Q(x)$;
$\neg \forall y(\exists x P\ (x, y) \to \forall u(R(y, u) \to \neg \forall z(P\ (z, u) \vee \neg R(z, y))))$;
$\exists x \exists y(P(x,y) \to R(x)) \to \forall x(\neg \exists y P(x,y) \vee R(x))$;
$\exists x \forall y\ (P(x,y) \to (\neg P(y,x) \to (P(x,x) \equiv P(y,y))))$;
$\exists x(\forall x P(x,x) \vee \exists x \neg R(x)) \to \exists x(R(x) \to \exists y P(f(x),y))$.

Упражнение 2. Построить сколемовские стандартные формы для указанных ниже формул:
$\forall x \exists y \forall z \exists u R(x, y, z, u)$;
$\neg \forall x(\exists y R(x, y) \to \forall z P\ (z, x))$;
$\neg \forall y(\exists x P\ (x, y) \to \forall u(R(y, u) \to \neg \forall z(P\ (z, u) \vee \neg R(z, y))))$;
$\exists x \exists y(P(x,y) \to R(x)) \to \forall x(\neg \exists y P(x,y) \vee R(x))$;
$\exists x \forall y\ (P(x,y) \to (\neg P(y,x) \to (P(x,x) \equiv P(y,y))))$;
$\exists x(\forall x P(x,x) \vee \exists x \neg R(x)) \to \exists x(R(x) \to \exists y P(f(x),y))$.

Глава 5. Метод резолюций

Без ограничения общности далее полагаем, что формула ϕ, общезначимость которой проверяется методом резолюций, замкнута.

5.1. Общая схема метода резолюций

Сквозной пример: обоснование общезначимости формулы
$\exists x\ (P(x)\ \&(\forall x\ P(x) \to \exists y\ R(x, y)) \to \exists y\ R(x, y))$
Этап 1: перейти к проверке невыполнимости отрицания формулы
$\neg \exists x\ (P(x) \& (\forall x\ P(x) \to \exists y\ R(x,y)) \to \exists y\ R(x,y))$
Этап 2: упростить формулу, сохранив её смысл (привести к ПНФ-предваренной нормальной форме)
$\forall x\ \exists z\ \exists y\ \forall u\ (P(x)\ \&(\neg P(z) \vee R(x, y))\ \&\ \neg R(x, u))$

Этап 3: сделать формулу ещё проще с изменением смысла, сохранив её выполнимость/невыполнимость (привести к ССФ- сколемовской стандартной форме)

$\forall x\ \forall u\ (P(x)\ \&(\neg P(f\,(x)) \vee R(x, g(x)))\ \&\ \neg R(x, u))$

Этап 4: перейти к проверке невыполнимости системы очень простых формул — дизъюнктов

$\{P(x), \neg P(f\,(x)) \vee R(x, g(x)), \neg R(x, u)\}$

Этап 5: построить логический вывод особого дизъюнкта $\square$, обозначающего невыполнимость системы.

Общая схема метода резолюций: исходная формула $\phi \mapsto$ отрицание $\psi = \neg\phi \mapsto$ ПНФ $\psi pnf \mapsto$ ССФ $\psi ssf \mapsto$ система дизъюнктов $S_{\psi ssf} \mapsto$ резолютивный вывод пустого дизъюнкта из $S_{\psi_{ssf}}$

Отдельно была показана справедливость цепочки равносильностей:

$\models \phi \Leftrightarrow \not\models\psi \Leftrightarrow \not\models \psi_{pnf} \Leftrightarrow \not\models \psi_{ssf} \Leftrightarrow \not\models S_{\psi_{ssf}} \Leftrightarrow$ существует вывод $\square$ из $S_{\psi_{ssf}}$

На последнем этапе метода резолюций

требуется проверить выполнимость системы дизъюнктов. Попробуем, по аналогии с методом семантических таблиц, придумать способ извлечения явных противоречий из скрытых (неявных).

В качестве явного противоречия будем использовать пустой дизъюнкт $\square$.

Начнём издалека: приведём несколько примеров того, как можно было бы «разумно» и «просто» извлечь $\square$ из невыполнимой системы дизъюнктов.

$\Vdash \{P(x), \neg P(x)\}$ - явное противоречие:

$$\begin{array}{ccc} P(x) & & \neg P(x) \\ & \diagdown\quad\diagup & \\ & \square & \end{array}$$

$\Vdash \{\neg P(x), \neg Q(y), P(x) \vee Q(y)\}$ - не явное противоречие

$$\begin{array}{lcccc} & & \neg P(x) & & P(x) \vee Q(y) \\ & & & \diagdown\quad\diagup & \\ \text{явное противоречие} & \neg Q(y) & & Q(y) & \\ & & \diagdown\quad\diagup & & \\ & & \square & & \end{array}$$

Отсюда следует, что $\forall x\ \neg P(x), \forall x\ \forall y\ (P(x) \vee Q(y)) \models \forall y\ Q(y)$

Противоречия в система дизъюнктов $\not\models \{P(f(x),y), \neg P(u,g(v))\}$.

$P(f(x), y)$ $\neg P(u, g(v))$ -не очень явное противоречие.

$\downarrow$ $\downarrow$

$P(f(x), g(v))$ $\neg P(f(x), g(v))$ - явное противоречие

\ /

$\square$

так как $\forall x\ \forall y\ P(f(x),y) \models \forall x\ \forall v\ P(f(x),g(v))$ и
$\forall u\ \forall v\ \neg P(u, g(v)) \models \forall x\ \forall v\ \neg P(f(x), g(v))$.

Чтобы обнаружить не очень явное противоречие в системе дизъюнктов, а затем явное, потребовалось привести атомы дизъюнктов к общему виду. Приведение выражений к общему виду называется унификацией, и описание последнего этапа метода резолюций (резолютивного вывода) необходимо начать со строгой формулировки задачи унификации и алгоритма ее решения.

5.2. Композиция подстановок

Унификация атомов A, B достигается применением к ним подстановки θ, такой что $A\theta = B\theta$.

Напоминание. Подстановка — это отображение θ : Var → Term. Конечная подстановка задаётся множеством связок: $\{x_1/t_1, ..., x_n/t_n\}$

$E\theta$ — это результат применения подстановки θ к выражению E.
Чтобы поставить и решить задачу унификации, исследуем основные алгебраические свойства подстановок.
Композиция подстановок θ и η — это подстановка $\theta\eta$, такая что для любой переменной x верно равенство.
$x(\theta\eta) = (x\theta)\eta$

Утверждение.
Пусть $\theta = \{x_1/t_1,...,x_n/t_n\}$ и $\eta = \{y_1/s_1,...,y_k/s_k\}$.
Тогда $\theta\eta=\{x_i/t_i\eta | 1\le i \le n, x_i \ne t_i\eta\}$ $\cup$ как это $\{y_j/s_j \mid 1 \le j \le k, y_j \notin \{x_1,...,x_n\}\}$

Доказательство.
Рассмотрим переменную $z \in Var$
Если $z \notin Dom_\theta \cup Dom_\eta$, то $z(\theta\eta) = (z\theta)\eta = z\eta = z$.
Если $z = y_j \in Dom_\eta \setminus Dom_\theta$, то $z(\theta\eta) = (z\theta)\eta = z\eta = s_j$

Иначе $z = x_i \in Dom_\theta$, и $z(\theta\eta) = (z\theta)\eta = t_i\eta$.

Пример.

Даны подстановки $\theta = \{x/f(x,c), y/g(u), z/y\}$ и $\eta = \{x/g(y), y/z, u/c\}$
Найти постановку $\theta\eta = ?$

$\{x/f(x,c)\eta, y/g(u)\eta, z/y\eta\} \cup \{u/c\} =$
$\{x/f(g(y),c), y/g(c), z/z\} \cup \{u/c\} = \{x/f(g(y),c), y/g(c)\} \cup \{u/c\}$
Таким образом мы нашли
$\theta\eta = \{x/f(g(y), c), y/g(c), u/c\}$

5.3. Задача унификации. Наиболее общий унификатор

Подстановка θ называется унификатором выражений E_1, E_2, если $E_1\theta = E_2\theta$.
У(E_1,E_2) — множество всех унификаторов выражений E_1, E_2
Выражения E_1, E_2 унифицируемы, если У$(E_1, E_2) \neq \varnothing$
Утверждение. Для любых подстановок θ, η и любых выражений E_1, E_2 верно: если $\theta \in$ У(E_1, E_2), то $\theta\eta \in$ У(E_1, E_2)
Доказательство.
$\theta \in$ У$(E_1,E_2) \Leftrightarrow E_1\theta = E_2\theta \Rightarrow E_1\theta\eta = E_2\theta\eta \Leftrightarrow \theta\eta \in$ У(E_1,E_2).
Подмножество S множества подстановок Θ называется полным в Θ, если любая подстановка θ из Θ представима в виде $\theta = \eta\mu$, где $\eta \in S$.

Подстановка θ называется *наиболее общим унификатором* выражений E_1, E_2, если множество $\{\theta\}$ является полным в У(E_1,E_2).
НОУ(E_1,E_2) — множество всех наиболее общих унификаторов выражений E_1, E_2
Пример. Рассмотрим два атома: $A = P(f(x), y)$ и $B = P(u, g(v))$.
Подстановка $\eta = \{y/g(g(v)), u/f(c), v/g(v), x/c\}$ — унификатор атомов A и B (то есть $\eta \in$ У(A,B)), т.к.
$P(f(x), y)\eta = P(f(c), g(g(v))) = P(u, g(v))\eta$.
А подстановка $\theta = \{y/g(v), u/f(x)\}$ — более общий унификатор A и B:
$P(f(x), y)\theta = P(f(x), g(v)) = P(u, g(v))\theta$
$\eta = \theta\{v/g(v), x/c\}$
На самом деле θ — наиболее общий унификатор атомов A и B.
(А как это доказать?)
А выражения $P(x,f(x))$ и $P(g(y),y)$ неунифицируемы.

(А это как доказать?)

Точная формулировка задачи унификации: для заданных выражений E_1, E_2 выяснить, унифицируемы ли эти выражения, и если это так, то вычислить множество унификаторов, полное в $У(E_1, E_2)$.

Чтобы освоить метод резолюций, достаточно научиться решать эту задачу для произвольной пары атомов.

Переход от атомов к системам уравнений

Утверждение. Никакая пара атомов $P(t_1,...,t_k)$, $Q(s_1,\dots,s_m)$ с различными предикатными символами $P^{(k)}$, $Q^{(m)}$ не унифицируема.

Доказательство. Очевидно.

Далее рассматривается только унификация атомов, содержащих одинаковые предикатные символы.

5.4. Система уравнений над термами

Переход от атомов к системам уравнений.

Унификация атомов $P(t_1,...,t_k)$, $P(s_1,...,s_k)$ $\Leftrightarrow$

Вычисление подстановки θ, такой что $t_1\theta = s_1\theta, \dots, t_k\,\theta = s_k\,\theta$ $\Leftrightarrow$

Вычисление подстановки θ, такой что левая и правая части каждого уравнения в системе $\mathcal{E}$ вида:

t1 =s1

…

tk =sk

становятся посимвольно одинаковыми при применении θ.

Вычисление решения системы уравнений $\mathcal{E}$ в свободной алгебре термов означает следующее: значение терма — это сам терм, то есть термы равны, если они посимвольно совпадают и операция композиции — это подстановка терма на место переменной.

Чтобы отличать равенство в уравнениях системы от посимвольного совпадения выражений от других видов равенства, будем знак равенства в системах уравнений записывать так: $\doteq$.

Системой уравнений (в свободной алгебре термов) будем называть запись $\mathcal{E}$ вида

$t_1 \doteq s_1$

... ,

$t_k \doteq s_k$

где $t_1,\ldots,t_k$, $s_1,\ldots, s_k$ — термы.

Подстановка θ — унификатор (решение) системы $\mathcal{E}$, если для каждого i, $i \in \{1,2,...,k\}$, верно $t_i\theta = s_i\theta$.

$У(\mathcal{E})$ — множество всех унификаторов системы уравнений $\mathcal{E}$.

Система уравнений $\mathcal{E}$ унифицируема (имеет решение), если $У(\mathcal{E})\neq\varnothing$.

$НОУ(\mathcal{E})$ — множество всех наиболее общих унификаторов системы уравнений $\mathcal{E}$.

Примеры.

Пусть $\mathcal{E}$: $f(c,x) \doteq f(y,g(y))$

$g(y) \doteq z$

и $\theta = \{x/g(c), y/c, z/g(c)\}$

Тогда $\mathcal{E}\theta$: $ff(c,g(c)) \doteq f(c,g(c))$

$g(c) \doteq g(c)$

Видно, что θ — унификатор системы (А как его вычислить и проверить, наиболее общий ли он?)

А система: $f(c,y) \doteq f(y,g(y))$

$g(y) \doteq z$

неунифицируема (А как такое доказать в общем случае?).

Утверждение. Множества унификаторов любой пары атомов $P(t_1,...,t_k)$, $P(s_1,\ldots,s_k)$

и системы уравнений $t1 \doteq s1$

...

$t_k \doteq s_k$

совпадают.

<u>Унификация переменной и терма.</u>

Лемма (о связке). Для любых переменной x и терма t верно:

1. Если $x = t$, то $\varepsilon\in НОУ(x,t)$
2. Если $x \notin Var_t$, то $\{x/t\} \in НОУ(x,t)$
3. Если $x\in Var_t$ и $x\neq t$, то $У(x,t)=\varnothing$

Доказательство.

1. Следует из того, что $x\varepsilon = x$ и для любой подстановки η верно

$\eta = \varepsilon\eta$

2. Достаточно показать, что если $x \notin Var_t$, то:

а) $\{x/t\}$ — унификатор (переменной x и терма t)

б) для любого унификатора θ существует подстановка η, такая что $\theta = \{x/t\}\eta$.

2а) Следует из равенств $x\{x/t\} = t = t\{x/t\}$

2б) $x \notin Var_t$ и $x\theta = t\theta \quad \Rightarrow \exists\eta\ \theta = \{x/t\}\eta$

Достаточно показать, что $\theta = \{x/t\}\theta$.

Для этого рассмотрим произвольную переменную y и покажем, что $y\theta = y\{x/t\}\theta$:

- Если y=x, то $y\theta=x\theta=t\theta=x\{x/t\}\theta=y\{x/t\}\theta$.
- Иначе $y \neq x$, и $y\theta=y\{x/t\}\theta$.

Следовательно, для любой переменной y верно равенство $y\theta = y\{x/t\}\theta$, а значит, верно и $\theta = \{x/t\}\theta$.

3. Рассмотрим произвольную подстановку θ и покажем, что если $x \in Var_t$ и $x \neq t$, то θ не может являться унификатором x и t.

Пусть $x\theta = s$

Тогда $|x\theta| = |s| < |t\{x/s\}| \leq |t\theta|$

Следовательно, $|x\theta| < |t\theta|$, а значит, $x\theta \neq t\theta$ ($|p|$ — длина терма p).

<u>Унификация приведённой системы.</u>

Система уравнений является приведённой, если она имеет вид:

$x_1 \doteq t_1$

... ,

$x_k \doteq t_k$

где $x_1, \dots, x_k$ — попарно различные переменные, не встречающиеся в правых частях уравнений.

Пример 1.

$x \doteq f(y,g(y))$

$z \doteq w$ — приведённая система

$u \doteq g(c)$

Пример 2.

$x \doteq f(y,g(y))$

$x \doteq w$.

$y \doteq g(c,c)$

$g(z) \doteq f(c, x)$ — неприведённая система:

- в левой части уравнения находится терм g (z);
- x встречается в левых частях два раза;

- y встречается и в левой, и в правой частях.

Лемма (о приведённой системе)

Если $\mathcal{E}$:

$$\begin{matrix} x_1 \doteq t_1 \\ \dots \\ x_k \doteq t_k \end{matrix} \quad \text{— приведённая система,}$$

то $\{x_1/t_1,\dots,x_k/t_k\} \in$ НОУ($\mathcal{E}$)

Доказательство. Следует из леммы о связке.

5.5. Унификация произвольной системы термов. Алгоритм Мартелли-Монтанари

Алгоритм, о котором будет дальше идти речь, коротко описывается так: это метод исключения переменных, широко применяющийся для решения систем линейных алгебраических уравнений и адаптированный к свободной алгебре термов.

Алгоритм унификации (Martelli A., Montanari U.) Далее будут описаны 6 правил преобразования системы уравнений. Эти правила произвольно (недетерминированно) применяются к системе, пока возможно.

Triv: Удалить $t \doteq t$

Swap: заменить $t \doteq x$ на $x \doteq t$, если $t \notin Var$

Func: заменить $f(t_1,\dots,t_k) \doteq f(s_1,\dots,s_k)$ на

$t_1 \doteq s_1$

…

$t_k \doteq s_k$

Eim: если в системе содержится уравнение Eq : $x \doteq t$, где $x \notin Var_t$ и x встречается в других уравнениях системы, то применить подстановку $\{x/t\}$ ко всем уравнениям системы, кроме Eq.

NElim: Явная неунифицируемость: ($x \in Var$, $t \in Term$) если в системе содержится уравнение $x \doteq t$, где $x \in Var_t$ и $x \neq t$,то

СТОП: система не унифицируема.

NFunc: если в системе содержится уравнение $f(t_1,\dots,t_k) \doteq g(s_1,\dots,s_m)$, где $f \neq g$, то

СТОП: система не унифицируема.

Унификация произвольной системы. Примеры.

Примеры:

1. $\mathcal{E}0$: $f(x,g(y)) \doteq f(g(y),x)$ $\mapsto$ Func

$c \doteq y$

$\mathcal{E}1$: $x \doteq g(y)$

$g(y) \doteq x$

$c \doteq y$ $\mapsto$ Swap

$\mathcal{E}2$: $x \doteq g(y)$

$g(y) \doteq x$

$y \doteq c$ $\mapsto$ Elim 2 раза

$\mathcal{E}3$: $x \doteq g(c)$

$g(c) \doteq g(c)$

$y \doteq c$ $\mapsto$ Triv

$\mathcal{E}4$: $x \doteq g(c)$

$y \doteq c$ приведённая система.

Ответ: $\{x/g(c), y/c\} \in$ НОУ(E).

2. $\mathcal{E}0$: $f(x, g(x)) \doteq h(g(y), x)$

$c \doteq y$ $\mapsto$ NFunc СТОП

Ответ: У($\mathcal{E}0$) $=\varnothing$

3. $\mathcal{E}0$: $f(x,g(x)) \doteq f(g(y),x)$

$c \doteq y$ $\mapsto$ Func

$\mathcal{E}1$: $x \doteq g(y)$

$g(x) \doteq x$

$c \doteq y$ $\mapsto$ Swap

$\mathcal{E}2$: $x \doteq g(y)$

$x \doteq g(x)$

$c \doteq y$ $\rightarrow$ NElim СТОП

Ответ: У($\mathcal{E}0$) $= \varnothing$

Теорема (об унификации). Для любой системы уравнений $\mathcal{E}_0$

1. алгоритм A завершает работу на $\mathcal{E}_0$ (завершаемость);

2. по завершении алгоритмом A выдаётся подстановка или сообщение СТОП (успешность);

3. если выдана подстановка θ, то $\theta \in$ НОУ($\mathcal{E}_0$) (корректность);

4. если выдано сообщение СТОП, то система $\mathcal{E}0$ неунифицируема (полнота).

Следствие. Для любых атомов A и B логики предикатов верно: У(A,B) $\neq \varnothing$ $\Leftrightarrow$ НОУ(A,B) $\neq \varnothing$

Ещё немного определений.
Положительная литера — это атом.
Отрицательная литера — это отрицание атома.
- Если E — логическое выражение и θ — подстановка, то: $E\theta$ — пример выражения E.
- Если $Var_{E\theta} = \varnothing$, то $E\theta$ — основной пример.
- Если θ:Var→Var—биекция, то θ — переименование, а $E\theta$ — вариант выражения E.

Пример.
Рассмотрим выражение $E = P(x, f(y)) \vee \neg R(y, c)$ и подстановки
$\theta = \{x/u, y/z, u/x, z/y\}$
$\eta = \{x/g(d), y/z\}$
$\mu = \{z/c\}$
$\varepsilon = \{\}$
Тогда:
- подстановки θ и ε — переименования, а η и μ — нет;
- $E\eta = P(g(d), f(z)) \vee \neg R(z, c)$ — пример выражения E.

5.6. Примеры применения правила резолюций

Правило резолюции

$$\frac{D_1 \vee L_1, \qquad D_2 \vee \neg L_2}{(D_1 \vee D_2)\theta}$$

Здесь
- D_1, D_2 — дизъюнкты
- L_1, L_2 — положительные литеры
- $\theta \in$ НОУ(L_1,L_2)

При использовании правила резолюции допускается перестановка слагаемых дизъюнктов.

Пример 1.

Контрарная пара: $P(x, f(y))$ и $\neg P(g(z, y), z)$ в дизъюнктах

$\mathbf{P(x, f(y))} \vee \neg R(g(x, z), f(z))$ и $Q(x) \vee R(y, x) \vee \mathbf{\neg P(g(z, y), z)}$.

Тогда

$\neg R(g(g(f(y),y), f(y)), f(f(y))) \vee Q(g(f(y), y)) \vee R(y, g(f(y), y))$ - резольвента этих дизъюнктов.

$\theta=\{x/g(f(y),y), z/f(y)$ $\in$ НОУ$(P(x,f(y)), P(g(z,y),z))$ и резольвента получена так:

$(\neg R(g(x, z), f(z)) \vee Q(x) \vee R(y, x))\theta$.

Пример 2.

Другая контрарная пара $\neg R(g(x, z), f(z))$ и $R(y, x)$ в тех же дизъюнктах

$P(x, f(y)) \vee \mathbf{\neg R(g(x, z), f(z))}$ и $Q(x) \vee \mathbf{R(y, x)} \vee \neg P(g(z, y), z)$.

Тогда

$P(f(z), f(g(f(z), z))) \vee Q(f(z)) \vee \neg P(g(z, g(f(z), z)), z)$ - резольвента этих дизъюнктов.

$\theta=\{x/f(z), y/g(f(z),z)\} \in$ НОУ$(R(g(x,z), f(z)), R(y,x))$ и резольвента получена так:

$(P(x, f(y)) \vee Q(x) \vee \neg P(g(z, y), z))\theta$.

Пример 3.

Контрарная пара $P(f(x), y)$ и $\neg P(u, g(v))$ в дизъюнктах

$P(f(x), y)$ и $\neg P(u, g(v))$.

$\square$ - резольвента.

$\theta = \{y/g(v), u/f(x)\} \in$ НОУ$(P(f(x), y), P(u, g(v)))$ и резольвента получена так: (пустое множество литер)θ.

5.7. Лемма (о корректности правила резолюции)

Если D — резольвента дизъюнктов D_1, D_2, то $D_1, D_2 \models D$

5.8. Правило склейки. Лемма о корректности правила склейки

Применение одного только правила резолюции далеко не всегда позволяет вывести □ из невыполнимой системы.

Например:

$\{P(x) \vee P(c),\ \neg P(c) \vee \neg P(y)\}$

Система из таких двух дизъюнктов невыполнима, но все резольвенты, резольвенты резольвент и т.д. этих дизъюнктов имеют ровно две литеры.

Необходимо иметь правило,

которое позволит получать □ и из таких систем.

Правило склейки

$$\frac{D \vee L_1 \vee L_2}{(D \vee L_1)\theta}$$

Здесь D — дизъюнкт, L_1, L_2 — литеры, $\theta \in НОУ(L_1, L_2)$

При использовании правила склейки допускается перестановка слагаемых дизъюнктов.

Дизъюнкт $(D \vee L_1)\theta$ — склейка дизъюнкта $D \vee L_1 \vee L_2$

Литеры L_1, L_2 образуют склеиваемую пару.

Пример. Склеиваемая пара в дизъюнкте

$$\frac{P(x) \vee \mathbf{\neg R(y, z, f(x))} \vee \mathbf{\neg R(x, f(c), z)}}{P(c) \vee \neg R(c, f(c), f(c))} \text{ - склейка}$$

$\theta = \{x/c, y/c, z/f(c)\} \in НОУ(\neg R(y, z, f(x)), \neg R(x, f(c), z))$ и склейка получена так: $(P(x) \vee \neg R(y, z, f(x)) \vee \neg R(x, f(c), z))\,\theta$

Лемма (о корректности правила склейки).

Если D — склейка дизъюнкта D_1, то $D_1 \models D$

5.9. Резолютивный вывод

Пусть S — система дизъюнктов.

Резолютивный вывод из S — это конечная последовательность дизъюнктов $D_1, ..., D_i, ..., D_k$,

такая что каждый дизъюнкт D_i является

- вариантом дизъюнкта из S,

- склейкой дизъюнкта D_j , где $j < i$, или

- резольвентой дизъюнктов D_j, D_m, где $j < i$ и $m < i$

Дизъюнкт резолютивно выводим из S, если существует резолютивный вывод из S, оканчивающийся этим дизъюнктом.

Пример.

$S = \{ P(x), \neg P(f(x)) \vee R(x, g(x)), \neg R(x, u)\}$

Резолютивный вывод □ из S:

1. $P(x_1)$ - вариант дизъюнкта P(x) из
2. $\neg P(f(x_2)) \vee R(x_2, g(x_2))$ - вариант $\neg P(f(x)) \vee R(x, g(x))$ из S
3. $R(x_3, g(x_3))$ - резольвента 1 и 2
4. $\neg R(x_4, u_4)$ - вариант дизъюнкта $\neg R(x, u)$ из S
5. □ - резольвента 3 и 4

Следовательно, □ резолютивно выводим из системы S.

Другой пример

Резолютивный вывод □ из

S: $\{\neg P(f(x),z) \vee \neg P(y,y),\ P(x, f(y)) \vee R(y), \neg R(y)\}$.

1. $P(x_1, f(y_1)) \vee R(y_1)$ -вариант дизъюнкта из S
2. $\neg R(y_2)$ - вариант дизъюнкта из S
3. $P(x_3, f(y_3))$ - резольвента 1 и 2
4. $\neg P(f(x_4), z_4) \vee \neg P(y_4, y_4)$ вариант дизъюнкта из S
5. $\neg P(f(x_5), f(x_5))$ - склейка 4
6. □ - резольвента 3 и 5.

Возможность использования всевозможных вариантов дизъюнктов наряду с самими дизъюнктами в резолютивном выводе настолько же важна, насколько и возможность использования резольвент и склеек.

Например: $S = \{\neg P(x), P(f(x))\}$

НОУ$(P(x),P(f(x))) = \varnothing$

Значит, у этих дизъюнктов нет ни одной резольвенты.

При этом система S невыполнима:

у формул $\forall x\ \neg P(x)$ и $\forall x\ P(f(x))$ нет общих моделей.

К вариантам же дизъюнктов из S применимо правило резолюции:

1. $P(x_1)$
2. $P(f(x_2))$ $\{x_1/f(x_2)\} \in$ НОУ$(P(x_1), P(f(x_2)))$
3. □ - резольвента 1 и 2.

Корректность использования всевозможных вариантов дизъюнктов обеспечивается следующей равносильностью:

$\forall x\, \varphi \sim \forall y\, (\varphi\{x/y\})$, если y свободен для постановки в φ на место x.

Резолютивный вывод успешен, если он оканчивается пустым дизъюнктом □. Успешный резолютивный вывод также называется резолютивным опровержением.

Если исходная система дизъюнктов выполнима, то система, к которой добавлены все дизъюнкты вывода, также выполнима (это следует из лемм о корректности правил резолюции и склейки).

Если среди добавленных дизъюнктов есть тождественно ложный □, то значит расширенная система дизъюнктов невыполнима.

Полученное противоречие опровергает выполнимость исходной системы (доказывает невыполнимость методом «от противного»).

Примеры, которые использовались при обсуждении этапов метода резолюций, выбирались так, чтобы при их совмещении получился сквозной пример: обоснование общезначимости формулы

$\exists x\, (P(x)\, \&(\forall x\, P(x) \to \exists y\, R(x, y)) \to \exists y\, R(x, y))$

методом резолюций.

Запишем получившееся обоснование от начала и до конца

Этап 1: поставить отрицание

$\models \exists x\, (P(x)\, \&(\forall x\, P(x) \to \exists y\, R(x, y)) \to \exists y\, R(x, y))$

$\Leftrightarrow$

$\not\Vdash\ \neg\exists x\, (P(x)\, \&(\forall x\, P(x) \to \exists y\, R(x, y)) \to \exists y\, R(x, y))$

Этап 2: построить равносильную ПНФ

$\neg\exists x\, (P(x)\, \&(\forall x\, P(x) \to \exists y\, R(x, y)) \to \exists y\, R(x, y))$

~ (переименование переменных)

$\neg\exists x\, (P(x)\, \&(\forall z\, P(z) \to \exists y\, R(x, y)) \to \exists u\, R(x, u))$ ~ (удаление импликаций)

$\neg\exists x\, (\neg(P(x)\, \&(\neg\forall z\, P(z) \vee \exists y\, R(x, y))) \vee \exists u\, R(x, u))$ ~ (продвижение отрицаний)

$\forall x\, (P(x)\, \&(\exists z\, \neg P(z) \vee \exists y\, R(x, y))\, \&\, \forall u\, \neg R(x, u))$ ~ (вынесение кванторов)

$\forall x\, \exists z\, \exists y\, \forall u\, (P(x)\, \&(\neg P(z) \vee R(x, y))\, \&\, \neg R(x, u))$ ~ (получение КНФ)

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

$\models \exists x\, (P(x)\, \&(\forall x\, P(x) \to \exists y\, R(x, y)) \to \exists y\, R(x, y)) \Leftrightarrow$

$\not\Vdash \forall x\, \exists z\, \exists y\, \forall u\, (P(x)\, \&(\neg P(z) \vee R(x, y))\, \&\, \neg R(x, u))$.

Этап 3: построить ССФ, применив алгоритм сколемизации

$\not\Vdash\ \forall x\, \exists z\, \exists y\, \forall u\, (P(x)\, \&(\neg P(z) \vee R(x, y))\, \&\, \neg R(x, u))$

$\Leftrightarrow$

$\not\models \forall x\, \forall u\, (P(x)\ \&(\neg P(f\,(x)) \vee R(x, g(x)))\ \&\ \neg R(x, u))$

Этап 4: перейти к системе дизъюнктов

$\not\models \forall x\, \forall u\, (P(x)\ \&(\neg P(f\,(x)) \vee R(x, g(x)))\ \&\ \neg R(x, u))$

$\Leftrightarrow$

$\not\models \{P(x), \neg P(f\,(x)) \vee R(x, g(x)), \neg R(x, u)\}$

Этап 5: Резолютивный вывод □ из

$S = \{P(x), \neg P(f\,(x)) \vee R(x, g(x)), \neg R(x, u)\}$

1. $P(x_1)$ - вариант дизъюнкта из S
2. $\neg P(f(x_2)) \vee R(x_2, g(x_2))$ - вариант дизъюнкта из S
3. $R(x_3, g(x_3))$ - резольвента 1 и 2
4. $\neg R(x_4, u_4)$ - вариант дизъюнкта из S
5. □ - резольвента 3 и 4

Оказалось, что □ резолютивно выводим из построенной системы дизъюнктов.

Следовательно (по совокупности доказанных ранее теорем), исходная формула $\exists x\, (P(x)\ \&(\forall x\, P(x) \to \exists y\, R(x, y)) \to \exists y\, R(x, y))$ общезначима.

5.10 Теорема о корректности резолютивного вывода

Теорема (о корректности резолютивного вывода).
Если из системы дизъюнктов S резолютивно выводим □, то система S невыполнима.

5.11. Теорема о полноте резолютивного вывода

Теорема. Из любой невыполнимой системы дизъюнктов резолютивно выводим пустой дизъюнкт.

5.12. Примеры применения метода резолюций

Пример 1. Проверить правильность рассуждения.
Некоторые пациенты любят своих докторов (F_1). Ни один пациент не любит знахаря (F_2) Значит, ни один доктор — не знахарь (G).
Решение. Переведем посылки и заключения на формальный язык:

$F_1 \Leftrightarrow \exists x(P\,(x)\ \&\ \forall y(D(y) \to L(x, y)))$,

$F_2 \Leftrightarrow \forall x(P\,(x) \to \forall y(Q(y) \to \neg\, L(x, y)))$,

$G \Leftrightarrow \forall x(D(x) \to \neg\, Q(x))$.

Проверяемое рассуждение имеет вид F_1 , $F_2 \vDash G$. Его справедливость эквивалентна невыполнимости формулы F_1 & F_2 & ¬ G.

Построим систему дизъюнктов.

Из F_1

(1) P (a)

(2) ¬ D(y) ∨ L(a, y)

Из F_2

(3) ¬P(x) ∨ ¬Q(y) ∨ ¬L(x,y)

Из ¬G

(4) D(b)

(5) Q(b)

Методом резолюций получается следующий вывод пустого дизъюнкта:

- (6) L(a, b) резольвента (4) и (2)
- (7) ¬ Q(y) ∨ ¬ L(a, y) резольвента (3) и (1)
- (8) ¬ L(a, b) резольвента (7) и (5)
- (9) □ резольвента (6) и (8)

Рассуждение справедливо.

Пример 2. Проверить правильность следствия

$$\forall x\, \exists y\, A(y, x)\ \&\ \forall x\, \forall y\, \forall z(A(x, y)\ \&\ A(x, z) \to A(y, z)) \vDash \forall x\, \forall y(A(x, y) \to A(y, x))$$

Обозначим:

F1=∀x ∃y A(y, x)

F2= ∀x ∀y ∀z(A(x, y) & A(x, z) → A(y, z)

G= ∀x ∀y(A(x, y) → A(y, x))

Построим систему дизъюнктов:

C1: A(f(x),x)	из F1
C2: ¬A(x,y) ∨ ¬A(x,z) ∨ A(y,z)	из F2
C3: A(a,b)	из ¬G
C4: ¬A(b,a)	из ¬G

Построим резолютивный вывод:

C5: ¬A(f(x),z) ∨ A(x,z)	резольвента C1,C2
C6: A(x,x)	резольвента C1,C5
C7: ¬A(a,z) ∨A(b,z)	резольвента C2,C3
C8: ¬A(a,a)	резольвента C4,C7
C9: □	резольвента C6,C8

Пример 3. Рассмотрим простой пример. Пусть брадобреи бреют всех людей, которые не бреются сами и не бреют тех, кто бреется сам. Тогда брадобреи не существуют.
Запишем это рассуждение на языке логики предикатов. Введем два предиката:
Бб(x) - x является брадобреем,
Б(x,y) - x бреет y.

F1: $\forall x$ Бб$(x)\rightarrow\forall y(\neg$Б$(y,y)\rightarrow$Б$(x,y))$

F2: $\forall x$ Бб$(x)\rightarrow\forall y($Б$(y,y)\rightarrow\neg$Б$(x,y))$

G: $\neg\exists x$ Бб(x)

После перевода в сколемовскую конъюнктивную нормальную форму получаем предложения:
$C1$: $\neg$Бб$(x)\vee$Б$(y,y)\vee$Б(x,y)

$C2$: $\neg$Бб$(x)\vee\neg$Б$(y,y)\vee\neg$Б(x,y)

$C3$: Бб(a)

Применяя метод резолюции и правило склейки, получаем предложения:
$C4$: $\neg$Бб$(x)\vee$Б(x,x) склейка $C1$

$C5$: $\neg$Бб$(x)\vee\neg$Б(x,x) склейка $C2$

$C6$: Б(a,a) резольвента $C3$ и $C4$

$C7$: $\neg$Б(a,a) резольвента $C3$ и $C5$

C8: □
Рассуждение справедливо.

Пример 4. Проверить правильность рассуждения.
Каждого, кто любит всех животных, кто-то любит (F1). Любого, кто убивает животных, никто не любит(F2). Джек любит всех животных(F3). Кота по имени Тунец убил либо Джек, либо Любопытство(F4). Действительно ли этого кота убило Любопытство(G)?

F1: $\forall x$ ($\forall y$ Животное$(y)\rightarrow$Любит$(x,y)) \rightarrow (\exists z$ Любит$(z,x))$

F2: $\forall x$ ($\exists y$ Животное$(y)\wedge$Убил$(x,y))\rightarrow (\forall z\ \neg$Любит$(z,x))$

F3: $\forall y$ Животное(y) → Любит(Джек,y)

F4: Кот(Тунец) ∧ (Убил(Джек,Тунец) ∨ Убил(Любопытство,Тунец))
F5: $\forall x$ Кот(x) → Животное(x)

G: Убил(Любопытство,Тунец)

Построим систему дизъюнктов:
$C1$: Животное$(f(x))$ ∨ Любит$(g(x),x)$

$C2$: ¬Любит(x,$f(x)$) ∨ Любит($g(x)$,x)
$C3$: ¬Животное(y) ∨ ¬Убил(x,y) ∨ ¬Любит(z,x)
$C4$: ¬Животное(y) ∨ Любит(Джек,y)
$C5$: Кот(Тунец)
$C6$: Убил(Джек,Тунец) ∨ Убил(Любопытство,Тунец)
$C7$: ¬Кот(x) ∨ Животное(x)
$C8$: ¬Убил(Любопытство,Тунец)

Применим правила резолюции и склейки:

$C9$: Животное(Тунец)	резольвента $C5$ и $C7$
$C10$: ¬Убил(x,Тунец) ∨ ¬Любит(z,x)	резольвента $C3$ и $C9$
$C11$: Убил(Джек,Тунец)	резольвента $C6$ и $C8$
$C12$: ¬Любит(z,Джек)	резольвента $C10$ и $C11$
$C13$: ¬Животное(f(Джек)) ∨ Любит (g(Джек),Джек)	резольвента $C2$ и $C4$
$C14$: Любит(g(Джек),Джек)	резольвента $C1$ и $C13$ и склейка
C15: □	резольвента $C12$ и $C14$

Таким образом, кота убило любопытство.

Пример 5. (Доказательство теоремы). Применим метод резолюций в доказательстве одной простой теоремы из теории групп.
В качестве исходной возьмем следующую аксиоматику теории групп:
F_1: ∀x, y, z ((xy)z=x(yz)) - ассоциативность умножения,
F_2: ∀x, y ∃z(zx=y) - существование решения уравнения zx=y,
F_3: ∀x, y ∃z(xz=y) - существование решения уравнения xz=y.

Предположим, что нам надо доказать теорему
G: ∃x∀y (yx=y), т.е. что в группе существует правая единица.
Наша задача – установить, что формула G есть логическое следствие формул F_1, F_2, F_3.

Прежде, чем решать эту задачу, перейдем к другой сигнатуре. Введем символ трехместного предиката P, который интерпретируется следующим образом: P(x,y,z) означает, что xy=z.

В новой сигнатуре формулы F_1, F_2, F_3 и G запишутся так:

$F_1 = \forall x,y,z,u,v,w\ (P(x,y,u) \& P(y,z,v) \& P(x,v,w) \supset P(u,z,w))$,

$F_2 = \forall x,y\ \exists z\ P(z,x,y)$,

$F_3 = \forall x,y\ \exists z\ P(x,z,y)$,

$G = \exists x\ \forall y\ P(y,x,y)$.

Сформируем множество $S=\{F_1,F_2,F_3,\neg G\}$, каждую из формул этого множества приведем к сколемовской нормальной форме и удалим кванторы общности. Получим множество дизъюнктов $\{D_1,D_2,D_3,D_4\}$:

$D_1= \neg P(x,y,u) \vee \neg P(y,z,v) \vee \neg P(x,v,w) \vee P(u,z,w)$,
$D_2= P(f(x,y),x,y)$,
$D_3= P(x,g(x,y)y)$,
$D_4=\neg P(h(x),x,h(x))$.

Построим вывод пустого дизъюнкта из множества дизъюнктов $D_1,...,D_4$.

Пусть эти дизъюнкты – начальные дизъюнкты вывода. Заменим переменные в дизъюнкте D_2, получим дизъюнкт $D_2 = P(f(x',y'),x',y')$.

Литералы P(x,y,u) из D_1 и $P(f(x',y'),x',y')$ из D_2' унифицируются подстановкой $s_1=\{x\backslash f(x',y'),y\backslash x', u\backslash y'\}$.

Применим правило резолюций к D_1 и D_2 (и указанным литералам), получим дизъюнкт

$D_5=\neg P(x',z,v) \vee \neg P(f(x',y')v,w) \vee P(y',z,w)$.

Далее, литерал $P(f(x',y'),v,w)$ из D5 и D_2 унифицируются подстановкой $s_2=\{x'\backslash x,y'\backslash y,v\backslash x,w\backslash y\}$.

Правило резолюций, примененное к D_5 и D_2, дает дизъюнкт

$D_6=\neg P(x,z,x) \vee P(y,z,y)$.

Резольвентой дизъюнктов D_3 и D_6 будет дизъюнкт

$D_7=P(y,g(y',y'),y)$.

Для получения этой резольвенты заменим переменные в D_3, получим $D_3=P(x',g(x',y'),y')$ и используем подстановку $s_3=\{x\backslash y',z\backslash g(y',y')\}$.

Наконец, из дизъюнктов D_4 и D_7 с помощью подстановки $s_4=\{y\backslash h(g(y',y')), x\backslash g(y',y')\}$ получаем

D8= □ - пустой дизъюнкт.

Пример 6. (Дедуктивные базы данных).

Отметим вначале одно свойство метода резолюций. Пусть сигнатура состоит из двух символов двухместных предикатов P и Q, которые интерпретируются следующим образом:

P(x,y) означает, что x– сын y,

Q(x,z) означает, что x– внук z.

Рассмотрим формулы:

$F_1=\forall x,y,z\ (P(x,y)\&P(y,z) \supset Q(x,z))$,

$F_2=\forall x\ \exists y P(x,y)$,

$G=\forall x\ \exists z Q(x,z)$,

смысл которых достаточно ясен.

Используя метод резолюций, покажем, что G есть логическое следствие F_1 и F_2. Приведем формулы $F_1,F_2,\neg G$ к сколемовской нормальной форме, получим дизъюнкты:

$D_1=\neg P(x,y) \vee \neg P(y,z) \vee Q(x,z)$,

$D_2=P(x,f(x))$,

$D_3=\neg Q(a,z)$.

Вывод пустого дизъюнкта получается довольно просто:

$D_4=\neg P(a,y) \vee \neg P(y,z)$ $\quad ((D_1\ D_3,)\{x=a\})$,

$D_5=\neg P(f(a),z)$ $\quad ((D_2\ D_4\ ),\{x=a,\ y=f(a)\})$,

$D_6=\square$ $\quad ((D_2\ D_5),\ \{x=f(a),\mathbf{z=f(f(a))}\}$.

Подстановка z=f(f(a)) означает, что дед элемента a есть отец отца элемента a. Таким образом, метод резолюций не только устанавливает факт логического следствия формулы G из формул F_1 и F_2, но еще и «подсказывает», как по данному x получить z такой, чтобы формула Q(x,z) была истинна.

Пример 7. (Планирование действий).

В качестве примера использования метода резолюций в задачах планирования действий рассмотрим известную в теории искусственного интеллекта задачу об обезьяне и бананах. В задаче говорится об обезьяне, которая хочет съесть бананы, подвешенные к потолку комнаты. Рост обезьяны недостаточен, чтобы достать бананы. Однако в комнате есть стул, встав на который обезьяна может достать бананы. Какие ей надо совершить действия, чтобы достать бананы?

Задачу формализуем следующим образом. Комнату с находящимися в ней обезьяной, стулом и бананами будем называть *предметной областью*. Конкретное местонахождение в комнате обезьяны, стула и бананов будем называть *состоянием предметной области.* Рассмотрим два предиката P(x,y,z,s) R(z). Пусть

$P(x,y,z,s)$ означает, что в состоянии s обезьяна находится в точке x, стул - в точке y, бананы – в точке z,

$R(s)$ означает, что в состоянии s обезьяна взяла бананы.

Возможности обезьяны формализуем следующим образом. Введем три функции, которые принимают значения в множестве состояний:

ИДТИ(x,y,s) – состояние, которое получится из s, если обезьяна из точки x перешла в y,

НЕСТИ(x,y,s) – состояние, которое получится из s, если обезьяна перенесла стул из точки x в y,

БРАТЬ(s) – состояние, которое получится из s, если обезьяна взяла бананы.

Условия задачи запишутся в виде следующих формул:

$F_1= \forall x,y,z,s\ (P(x,y,z,s) \supset P(y,y,z,\text{ИДТИ}(x,y,s))$,

$F_2=\forall x, u, s\ (P(x,x,u,s) \supset P(u,u,u, \text{НЕСТИ}(x,u,s))$,

$F_3=\forall x\ (P(x,x,x,s) \supset R(\text{БРАТЬ}(s)))$.

Пусть в начальном состоянии s_0 обезьяна находилась в точке a, стул–в точке b, бананы–в точке c. Следовательно, к написанным формулам надо добавить формулу

$F_4=P(a,b,c,s_0)$.

Надо показать, что формула $G=\exists sR(s)$ есть логическое следствие формул F_1,F_2,F_3,F_4.

Из множества формул F_1,F_2,F_3,F_4, $\neg G$ получим множество дизъюнктов D_1–D_5 (к дизъюнкту, полученному из $\neg G$ добавлен литерал ответа $ANS(s)$):

$D_1=\neg P(x,y,z,s) \vee P(y,y,z,\text{ИДТИ}(x,y,s))$,

$D_2=\neg P(x,x,u,s,) \vee P(u,u,u,\text{НЕСТИ}(x,u,s))$,

$D_3=\neg P(x,x,x,s) \vee R(\text{БРАТЬ}(s))$,

$D_4=P(a,b,c,s_0)$,

$D_5=\neg R(s) \vee ANS(s)$.

Последовательность дизъюнктов D_1–D_5 продолжаем до вывода литерала ответа:

$D6=\neg P(x,x,x,s) \vee ANS(\text{БРАТЬ}(s))$ (из D_3 D_5),

$D_7=\neg P(x,x,u,s) \vee ANS(\text{БРАТЬ}(\text{НЕСТИ}(x,u,s)))$ (из D_2 и D_6),

$D_8=\neg P(x,y,z,s) \vee ANS(\text{БРАТЬ}(\text{НЕСТИ}((y,z,\text{ИДТИ}(x,y,s))))$ (из D_1 и D_7),

$D_9=ANS(\text{БРАТЬ}(\text{НЕСТИ}(b,c,\text{ИДТИ}((a,b,s_0))))$ (из D_4 и D_8).

Итак, для того, чтобы обезьяне взять бананы, надо сначала из точки a идти в точку b, затем из точки b нести стул в точку c и в точке c, встав на стул, взять бананы.

5.13. Упражнения для самопроверки

1. Проверить справедливость логических следствий

1. $\models \forall x\ (P1(x) \rightarrow \neg P2(x)) \rightarrow \neg(\exists x\ (P1(x)) \& \forall x(P2(x)))$.
2. $\forall x(\ P1(x) \rightarrow (\neg\ P2(x)));\ \forall x(P3(x) \rightarrow P1(x))\ \models \forall x(P3(x) \rightarrow (\neg\ P2(x)))$.
3. $\forall y\ \ (P3(y) \& \neg P4(y)\ \ \rightarrow\ \ \forall x(P1(x) \& P2(x,y)));\ \ \exists y(P3(y) \& P5(y));\ \forall y(P3(y) \& P4(y) \rightarrow \neg P5(y))\ \models \exists x(P1(x) \& P5(x))$.
4. $\exists x(P1(x) \& \forall y(P2(y) \rightarrow P4(x;y)));\ \forall x(P1(x) \rightarrow \forall y(P3(y) \rightarrow \neg P4(x;y)))$
$\models \forall x(P2(x) \rightarrow \neg P3(x))$.
5. $\exists x(A(x) \rightarrow B(x))\ \&\ \forall x(\neg A(x) \rightarrow B(x))\ \&\ \forall x(\neg B(x)\ V\ C\ (x))\ \models \exists x\ C\ (x)$.
6. $\forall x \forall y(A(f\ (x),\ y) \vee B(x,\ y))\ \&\ \forall x \forall y(\neg A(x,\ g(y)) \vee B(x,\ y))\ \models \exists x \exists y B(x,\ y)$.
7. $\forall x \forall y(A(x,\ y) \vee B(y,\ x))\ \&\ \forall x \forall y(A(x,\ y) \rightarrow C\ (x)) \&$
$\exists x \forall y(B(x,\ y) \rightarrow C\ (x))\ \models \exists x C\ (x)$.
8. $\models \neg\ (\ \neg \forall x\ (P\ (x) \rightarrow \forall y\ (P\ (y) \rightarrow ((Q(x) \rightarrow \neg Q(y)) \vee$
$\forall z\ P\ (z))))\ )$.
9. $\forall x\ \forall y\ (P1(x,\ y) \rightarrow P2(x,\ y));\ \forall x\ \forall y\ (P2(x,\ y) \rightarrow P3(x,\ y))\ \models \neg\ \exists x\ \exists y\ P1(x,\ y)$.

2. Проверить правильность суждений

1. Преподаватели принимали зачеты у всех студентов, не являющихся отличниками. Некоторые аспиранты и студенты сдавали зачеты только аспирантам. Ни один из аспирантов не был отличником. Следовательно, некоторые преподаватели были аспирантами.
2. Существуют студенты, которые любят всех преподавателей. Ни один из студентов не любит невежд. Следовательно, ни один из преподавателей не является невеждой.
3. Некоторые республиканцы любят всех демократов. Ни один республиканец не любит ни одного социалиста. Следовательно, ни один демократ не является социалистом.
4. Ни один торговец наркотиками не является наркоманом. Некоторые наркоманы привлекались к ответственности. Следовательно, некоторые люди, привлекавшиеся к ответственности, не являются торговцами наркотиков.
5. Саша – мальчик, у которого нет машины. Таня –девочка, которая любит мальчиков, имеющих машины. Следовательно, Таня не любит Сашу.

5.14. Практические реализации метода резолюций

Можно заметить, что при практическом применении метода резолюций мы столкнемся с его недетерминированностью. На каждом шаге в полученном множестве дизъюнктов необходимо выбрать два дизъюнкта, содержащих контрарную пару литер, и применить к ним правило резолюции, или выбрать дизъюнкт, содержащий унифицируемые литеры и применить к нему правило склейки. Такой выбор дизъюнктов для применения правила резолюции или правила склейки может быть осуществлен разными способами, что приводит на практике к драматическому росту размера множества дизъюнктов, участвующих в поиске резолютивного вывода.

От способа выбора дизъюнктов для применения правил резолютивного вывода существенно зависит возможность вывести □.

Стратегия резолютивного вывода — это набор ограничений на выбор дизъюнктов, к которым применяются правила при построении вывода.

Стратегия резолютивного вывода называется полной, если для любой невыполнимой системы дизъюнктов S существует вывод □ из S, построенный согласно этой стратегии.

Например, стратегия, никак не ограничивающая выбор дизъюнктов, полна но очень неэффективна. Существуют и более «эффективные» полные стратегии.

Входная резолюция

Входной резолютивный вывод из системы S, инициированный дизъюнктом D этой системы, устроен так:

1. Нечётные дизъюнкты вывода (при нумерации с единицы) называются центральными, чётные — боковыми;
2. Первый дизъюнкт вывода — это D;
3. Все боковые дизъюнкты являются вариантами дизъюнктов из S;
4. Каждый центральный дизъюнкт, кроме первого, — это резольвента двух предшествующих дизъюнктов

Проще говоря, во входном выводе каждый следующий центральный дизъюнкт — это резольвента предыдущего центрального дизъюнкта и варианта дизъюнкта исходной системы.

Входная резолюция — это стратегия, согласно которой разрешено строить любые входные резолютивные выводы и только их.

Пример: один из входных резолютивных выводов из
$S=\{\neg A\vee\neg B\vee C, A\vee C, B\vee C, \neg C\}$ устроен так:

Центральные:	Боковые:	
1. $\neg C$	2. $\neg A \vee \neg B \vee C$	$\mapsto$
3. $\neg A\vee\neg B$	4. $A\vee C$	$\mapsto$
5. $\neg B \vee C$	6. $\neg C$	$\mapsto$
7. $\neg B$	8. $B\vee C$	$\mapsto$
9. C	10. $\neg C$	$\mapsto$

□

Теорема (о неполноте входной резолюции).
Входная резолюция неполна.

Доказательство.
Вот пример системы, из которой не существует входного резолютивного вывода:
$S = \{A \vee A, \neg A \vee \neg A\}$.
При построении входного вывода запрещено применять правило склейки, а значит, все дизъюнкты входного вывода из S содержат ровно две литеры. Запрет на применение правила склейки — не единственная причина неполноты входной резолюции.
Даже если добавить возможность склеивать дизъюнкт перед построением резольвенты, то (как можно показать полным перебором) невозможно вывести □, например, из невыполнимой системы
$\{A \vee C, \neg A \vee C, B \vee \neg C, \neg B \vee \neg C\}$.

Рассмотрим снова следующую задачу.
Дано:
1. Даша любит Сашу, $\phi_1 = L(Д, C)$
2. Саша любит пиво, $\phi_2 = L(C, п)$
3. Паша любит пиво и всех тех, кто любит то же, что и он,
$\phi_3 = L(П,п)$
$\psi_1 = \forall x\ (\exists y\ (L(П,y) \& L(x,y)) \rightarrow L(П,x))$
Выяснить, любит ли кто-нибудь Дашу
$\chi = \exists x\ L(x,Д)$.
Т.о. Необходимо проверить справедливость логического следствия
$\phi_1, \phi_2, \phi_3, \psi_1 \models \chi$? , и выяснить кто любит Дашу.
Или по теореме о логическом следствии необходимо проверить общезначимость формулы:
$\models \phi_1 \& \phi_2 \& \phi_3 \& \psi_1 \rightarrow \chi$.
Решим эту задачу методом резолюций.
$\phi_1 = L(Д,C)$
$\phi_2 = L(C,п)$
$\phi_3 = L(П,п)$
$\psi_1 = \forall x\ (\exists y\ (L(П,y) \& L(x,y)) \rightarrow L(П,x))$

$\chi = \exists x\, L(x,Д)$

Проверим общезначимость формулы $\models \phi_1 \& \phi_2 \& \phi_3 \& \psi_1 \rightarrow \chi$?

Этап 1: поставим отрицание над формулой

$\neg(\phi_1 \& \phi_2 \& \phi_3 \& \psi_1 \rightarrow \chi)$

Этап 2: построим равносильную ПНФ

$\forall x \forall y \forall z\, (L(Д,С) \& L(С,п) \& L(П,п) \&$

$\& (\neg L(П,y) \vee \neg L(x,y) \vee L(П,x)) \& \neg L(z,Д))$

Этап 3: построим равновыполнимую ССФ. Формула выше — уже ССФ.

Этап 4: перейдём к системе дизъюнктов.

$S = \{ L(Д,С), L(С,п), L(П,п), \neg L(П,y) \vee \neg L(x,y) \vee L(П,x), \neg L(z,Д)\}$

Этап 5: попробуем вывести пустой дизъюнкт по стратегии входной резолюции:

$\neg L(z,Д)$	$\neg L(П,y') \vee \neg L(x',y') \vee L(П,x')$	$\{z/П, x'/Д, y'/y\}$	$\mapsto$
$\neg L(П,y) \vee \neg L(Д,y)$	$L(Д,С)$	$\{y/С\}$	$\mapsto$
$\neg L(П,С)$	$\neg L(П,y') \vee \neg L(x',y') \vee L(П,x')$	$\{x'/С, y'/y\}$	$\mapsto$
$\neg L(П,y) \vee \neg L(С,y)$	$L(П,п)$	$\{y/п\}$	$\mapsto$
$\neg L(С,п)$	$L(С,п)$	ε	

$\square$

Это успешный входной резолютивный вывод, инициированный дизъюнктом $\neg L(z,Д)$. Значит, $\phi_1, \phi_2, \phi_3, \psi_1 \models \chi$, то есть кто-то действительно любит Дашу. А кто?

Выпишем преобразования центральных дизъюнктов:

$\neg L(z,Д)\ \ \theta_1=\{z/П,x'/Д,y'/y\} \mapsto$

$\neg L(П,y) \vee \neg L(Д,y)\ \theta_2=\{y/С\} \mapsto$

$\neg L(П,С)\ \theta_3=\{x'/С,y'/y\} \mapsto$

$\neg L(П,y) \vee \neg L(С,y)\ \theta_4=\{y/п\} \mapsto$

$\neg L(С,п)\ \ \theta_5=\varepsilon \mapsto\ \ \square$

Тайный поклонник Даши в выводе обозначен переменной z. Посмотрим, как эта переменная изменялась унификаторами: $z\theta_1\theta_2\theta_3\theta_4\theta_5 = П$ (Паша). Оказывается, что Дашу любит Паша (могут быть и другие поклонники, но про них мы ничего не знаем)

Ниже выясним, всегда ли работает этот «трюк» с применением подстановок.

Хорновские дизъюнкты

Правилом будем называть дизъюнкт, содержащий ровно одну положительную литеру, то есть имеющий вид

$\neg A_1 \vee \cdots \vee \neg A_k \vee B$.

Запросом будем называть дизъюнкт, не содержащий ни одной положительной литеры (в том числе □), то есть имеющий вид

$\neg A_1 \vee \cdots \vee \neg A_k$

Хорновскими дизъюнктами называются правила и запросы.

Утверждение. $\neg A_1 \vee \cdots \vee \neg A_k \vee B \sim A_1 \& ... \& A_k \to B$

Утверждение. $\neg A_1 \vee \cdots \vee \neg A_k. \sim \neg(A_1 \& ... \& A_k)$

Основываясь на этих утверждениях, дизъюнкты

$\neg A_1 \vee \cdots \vee \neg A_k \vee B$ и $\neg C_1 \vee \cdots \vee \neg C_k$ будем иногда записывать как формулы

$A_1 \& ... \& A_k \to B$ и $\neg(C_1 \& ... \& C_k)$

Утверждение. Контрарную пару с заданной литерой запроса может образовывать не более одной литеры правила

Утверждение. Ни к какой паре запросов нельзя применить правило резолюции.

Утверждение. Резольвента запроса и правила является запросом.

Правило $A_1 \& ... \& A_k \to B$ — это естественный способ представления причинно-следственных взаимосвязей: C'Если справедливы факты $A_1,...,A_k$, то справедлив и факт B'. То есть $A_1,...,A_k$ —достаточное условие справедливости B. Если k = 0, то правило просто означает, что «Справедлив факт B». Согласно правилу, чтобы решить задачу, записанную в виде B, достаточно решить задачи, записанные в виде $A_1,...,A_k$, и совместить ответы. Если k = 0, то «Решение задачи B уже известно».

Формула $A_1 \& ... \& A_k$, отрицанием которой является запрос, — это естественный способ представления вопроса к базе знаний:

- Требуется проверить справедливость набора (взаимосвязанных) фактов $A_1,...,A_k$
- Требуется решить набор задач, записанных в виде $A_1, . . . , A_k$

Например, система дизъюнктов

$S = \{ L(Д,С), L(С,п), L(П,п), \neg L(П,y) \vee \neg L(x,y) \vee L(П,x), \neg L(z, Д)\}$

из задачи о Даше, Саше, Паше и пиве — это система хорновских дизъюнктов:

В части «Дано» (в базе знаний) записаны правила:

$L(Д, С)$
$L(С, п)$
$L(П, п)$
$L(П,y) \& L(x,y) \to L(П,x)$
Вопрос задачи, записанный в виде формулы и дополненный отрицанием в результате преобразования к дизъюнктам, — это запрос: $\neg L(z, Д)$.
Теорема о входной резолюции как средстве вычисления ответов на запросы к БЗ.
Пусть S — система дизъюнктов-правил,
Q_1 — дизъюнкт-запрос,
$Q_1,D_1,Q_2,D_2,...,Q_k,D_k,$, $k \geq 0$, — успешный **входной** резолютивный вывод из $S \cup \{Q_1\}$,
$\theta_1, \ldots, \theta_k$ — наиболее общие унификаторы, согласно которым в выводе строятся резольвенты $Q_2,...,Q_k$, соответственно, и
$\mathrm{Var}_{Q_1\theta_1...\theta_k} = \{x_1, \ldots, x_n\}$.
Тогда $S \models \forall \tilde{x}^n ((\neg Q_1)\theta_1 ...\theta_k)$
По только что доказанной теореме, в задаче о Даше, Саше, Паше и пиве
$S \models L(z, Д)\theta_1\theta_2\theta_3\theta_4\theta_5$
То есть $S \models L(П, Д)$: Паша действительно любит Дашу.
Выясняется, что «трюк» с применением подстановок в случае с хорновскими базами знаний работает всегда и превращается в надежный метод вычисления ответов на запросы.

5.15. Несколько слов о языке Пролог

Изложенные выше результаты о применении метода резолюций для Хорновских формул стали основой разработки языка Пролог.
Программа на Прологе состоит из множества хорновских дизъюнктов, записываемых в виде

$$Q((\vec{t})) : -P_1(\vec{t}_1), \ldots, P_n(\vec{t}_n), .$$

Корме того, в программе имеется цель вида
$: -R_1,...,R_k.$

Каждый шаг выполнения программы состоит в преобразовании цели путем ее унификации с одним из дизъюнктов. При взаимодействии цели
$:-Q(t), R_1(s_1), \ldots, R_k(s_k)$
с дизъюнктом

$Q((\vec{t}^*)){:}{-}P_1(\vec{t}_1), \ldots, P_n(\vec{t}_n)$, получается дизъюнкт вида

$: -P_1(\vec{t}^*{}_1), \ldots, P_n(\vec{t}^*{}_n), R_1(s^*{}_1), \ldots, R_k(s^*{}_k).$

Программа считается успешно завершенной, если в некоторый момент из цели исчезают все предикаты. Программа может зафиксировать неудачу, если один из предикатов цели ни с одним из дизъюнктов программы не унифицируется. Естественно, что может быть и промежуточный, но гораздо чаще встречающийся случай: программа не может зафиксировать неудачу, а просто зацикливается либо переполняется из-за неограниченного удлинения выражений.

Эта схема могла быть реализована несколькими способам. Поскольку Пролог появился в самом начале 70-х гг., был выбран способ, тогда находившийся вполне на уровне, но сейчас уже устаревший. Выбирается всегда первый член целевого дизъюнкта и первый из унифицируемых с ним дизъюнктов программы.

Здесь возникла сложность, которая была удачно разрешена и составила одно из важнейших достижений Пролога. Взяв первого кандидата, мы можем через несколько шагов зайти в тупик, а решение было совсем рядом: надо было взять следующего. Тут работает механизм возвратов (backtracking). Если фиксируется неудача, мы возвращаемся к первой точке, где было несколько кандидатов на унификацию, и подставляем следующий из возможных дизъюнктов. Этот механизм явился красивой и экономичной с точки зрения представления программ альтернативой явному выписыванию условных операторов. Но, конечно же, с точки зрения исполнения программ он может сильно проигрывать в эффективности.

Правило «брать первого кандидата из не отвергнутых ранее» обладает и другими особенностями. Во-первых, логика перестает быть классической, поскольку тривиально истинный дизъюнкт вида

$P(x) : -P(x)$

при помещении в программу вполне может привести к ее зацикливанию (если он применится однажды, то он будет применяться бесконечно). Во-вторых (и это уже большой плюс), появляется возможность выражать циклы и индукцию при помощи дизъюнктов типа

$A(n + 1) : -A(n), B.$

Далее Пролог, просуществовав некоторое время в университетской среде, неожиданно получил громадную рекламу в связи с тем, что японцы объявили его внутренним языком своего проекта ЭВМ пятого поколения.

Сегодня сам термин *логическое программирование* понимается как программирование на Прологе.

Пример. Программа на языке Пролог для решения задачи о
о Даше, Саше, Паше и пиве:

Правила:
:-любит(даша, саша);
:-любит(саша, пиво);
:-любит(паша, пиво);
:- любит(паша, X) ← любит(паша, Y), любит(X, Y);
Запрос Q:
?любит(X, даша)
И Пролог найдет один ответ X=паша.
Имеется многочисленная литература на русском языке по языку Пролог, и поэтому нет смысла здесь его подробно излагать.

Литература

Основная

1. Клини С. Математическая логика. М.:Мир, 1973, 480 с.
2. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. М.:Мир, 1983. 360 с.
3. Лавров И.А., Максимова Л.Л. Задачи по теории множеств, математической логике и теории алгоритмов. Москва, "Физико-математическая литература", 1995 г., 250 с.
4. Братко И. Программирование на Прологе для искусственного интеллекта. М.:Мир, 1990, 560 с.
5. Набебин А.А. Логика и Пролог в дискретной математике. М., Изд-во МЭИ, 1997.
6. Захаров В.А. Лекции по математической логике и теории алгоритмов. https://mk.cs.msu.ru/index.php/Участник:ZakharovVA
7. Подымов В.В. Лекции по математической логике и логическому программированию. https://mk.cs.msu.ru/index.php/Математическая_логика

Дополнительная

1. Мендельсон Э. Введение в математическую логику. М.:Наука, 1984. 319 с.
2. Верещагин Н.К., Шень А. Языки и исчисления. 2004.
3. Успенский В.А., Верещагин Н.К., Плиско В.Е. Вводный курс математической логики. 2004. 128 с.
4. Колмогоров А.Н., Драгалин А.Г. Математическая логика. Серия "Классический университетский учебник". Изд.3, 2006, 240 с.
5. Непейвода Н. Н. Прикладная логика. Новосибирск. 2000 г.
6. Хоггер К., Введение в логическое программирование. М.:Мир, 1988. 348 с.
7. Стерлинг Л., Шапиро Э., Искусство программирования на языке ПРОЛОГ. Москва, "Мир", 1990, 235 с.
8. Ковальский Р. Логика в решении проблем. М.: Наука, 1990. 277 с.
9. Логический подход к искусственному интеллекту (от модальной логики к логике баз данных). М.:Мир, 1998. 495 с.