

МГТУ ФН-12 МГТУ ФН-12 МГТУ ФН-12

Московский государственный технический университет
имени Н.Э. Баумана

Факультет «Фундаментальные науки»
Кафедра «Математическое моделирование»

А.Н. Канатников

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Конспект лекций

Для студентов кафедры ИУ9

Москва
2009

МГТУ ФН-12 МГТУ ФН-12 МГТУ ФН-12

7. МЕТОД РЕЗОЛЮЦИЙ

Одно из прикладных направлений развития логики — автоматический поиск доказательств. Основу для соответствующих алгоритмов составляют формальные системы генценовского типа, в которых вывод секвенций обладает высокой алгоритмичностью. И хотя, как было доказано, не существует универсальной разрешающей процедуры для построения вывода любой формулы или ее опровержения, генценовский вывод показывает, что для многих формул конечная разрешающая процедура существует.

Мы рассмотрим один из подходов к решению поставленной проблемы — так называемый **метод резолюций**. Он приспособлен для формул исчисления предикатов (или высказываний) определенного вида. Оказывается, что любая формула может быть преобразована в эквивалентную форму, пригодную для применения метода резолюций.

7.1. Скулемовские функции

Напомним, что любая формула алгебры предикатов может быть приведена к предваренной форме, т.е. форме вида $\nabla x_1 \nabla x_2 \dots \nabla x_k X[x_1, x_2, \dots, x_k]$, где формула X (матрица предваренной формы) не имеет кванторов, а символ ∇ обозначает один из кванторов.

Формула X , как бескванторная, может быть преобразована в эквивалентную конъюнктивную нормальную форму. В исчислении высказываний это можно сделать через таблицы истинности, но более эффективно (а в исчислении предикатов и неизбежно) использовать метод эквивалентных преобразований. Достаточно выразить все импликации и эквиваленции через отрицание, дизъюнкцию и конъюнкцию, а затем преобразовать формулу по законам булевой алгебры, рассматривая конъюнкцию как сложение, а дизъюнкцию как умножение.

Следующий шаг — такое преобразование предваренной формы, при котором в ее префиксе нет кванторов существования. Делается это путем расширения языка с добавлением констант и функциональных символов следующим образом. Пусть кванторная приставка ∇x_r содержит квантор существования, т.е. имеет вид $\exists x_r$. Если левее ее в префиксе нет кванторов всеобщности, выбираем константу a_s , не содержащуюся в формуле X , в этой формуле заменяем все вхождения переменной x_r на константу c_r и кванторную приставку удаляем:

$$\exists x_1 \dots \exists x_{r-1} \exists x_r \nabla x_{r+1} \dots \nabla x_k X \Rightarrow \exists x_1 \dots \exists x_{r-1} \nabla x_{r+1} \dots \nabla x_k X_{c_r}^{x_r}.$$

Пусть $i_1, i_2, \dots, i_s$ — список номеров всех кванторов всеобщности, предшествующих кванторной приставке $\exists x_r$. Вводим новый функциональный символ f , тип которого определяется последовательностью сортов переменных $x_r, x_{i_1}, x_{i_2}, \dots, x_{i_s}$, затем в формуле X каждое вхождение переменной x_r заменяем термом $f(x_{i_1}, x_{i_2}, \dots, x_{i_s})$, а кванторную приставку $\exists x_r$ из префикса удаляем.

Добавляемые в описанной процедуре константы и функциональные символы называют **скулемовскими функциями**, а полученную формулу в предваренной форме без кванторов существования с матрицей в конъюнктивной нормальной форме — **скулемовской стандартной формой**.

Пример 7.1. Построим скулемовскую стандартную форму для формулы

$$\exists x \forall y \forall z \exists u \forall v \exists w P(x, y, z, u, v, w).$$

В префиксе формулы три квантора существования. Левее $\exists x$ нет кванторов всеобщности. Поэтому первый шаг — замена переменной x константой. Получим:

$$\forall y \forall z \exists u \forall v \exists w P(c_1, y, z, u, v, w).$$

Перед приставкой $\exists u$ два квантора существования. Поэтому переменную u меняем на двуместный функциональный символ:

$$\forall y \forall z \forall v \exists w P(c_1, y, z, f_1(y, z), v, w).$$

перед приставкой $\exists w$ три квантора всеобщности. Значит, используем трехместный функциональный символ:

$$\forall y \forall z \forall v p(c_1, y, z, f_1(y, z), v, f_2(y, z, v)).$$

Последняя формула и есть скелемовская стандартная форма.

Пример 7.2. Построим скелемовскую стандартную форму для формулы

$$\forall x \exists y \exists z ((\neg p(x, y) \wedge q(y, z)) \vee r(x, y, z)).$$

Первый шаг — приведение матрицы приведенной формы к КНФ:

$$\forall x \exists y \exists z ([\neg p(x, y) \vee r(x, y, z)] \wedge [q(y, z) \vee r(x, y, z)]).$$

Далее элиминируем (т.е. убираем) первый слева квантор существования:

$$\forall x \exists z ([\neg p(x, f_1(x)) \vee r(x, f_1(x), z)] \wedge [q(f_1(x), z) \vee r(x, f_1(x), z)]).$$

Наконец, убираем и второй квантор существования:

$$\forall x ([\neg p(x, f_1(x)) \vee r(x, f_1(x), f_2(x))] \wedge [q(f_1(x), f_2(x)) \vee r(x, f_1(x), f_2(x))]).$$

Чтобы объяснить смысл описанного преобразования формул, введем несколько понятий. **литера** — это элементарная формула алгебры предикатов или ее отрицание. Дизъюнкция нескольких литер называют **дизъюнктом**. Иногда дизъюнкт удобно интерпретировать как множество литер, а не их дизъюнкцию. При этом пустое множество литер представляет собой **пустой дизъюнкт**, который обозначают символом „ $\square$ “. Считают, что пустой дизъюнкт — всегда ложная формула. Далее будем рассматривать конечные множества дизъюнктов. Такое множество является формой представления конъюнкции всех дизъюнктов этого множества. Отсюда вытекает понятие противоречивого множества дизъюнктов как множества дизъюнктов, конъюнкция которых, замкнутая кванторами всеобщности, дает противоречие (т.е. формула, ложная в любой интерпретации).

В скелемовской стандартной форме матрицу как КНФ можно интерпретировать как множество нескольких дизъюнктов. Мы также считаем, что каждая свободная переменная связана квантором всеобщности. Суть преобразования к стандартной форме — в сохранении условия противоречивости.

Теорема 7.1. Пусть S — множество дизъюнктов из стандартной формы для формулы F . Тогда формула F противоречива в том и только в том случае, когда множество дизъюнктов S противоречиво.

◀ Мы сразу можем считать, что формула F дана в предваренной форме. В принципе убирать кванторы существования можно в любом порядке (эти операции независимы и отличаются лишь выбором свободных констант и функциональных символов). Но удобнее это делать в порядке слева направо. Более того, достаточно доказать, что условие противоречивости сохраняется при убирании одного квантора.

Пусть формула F имеет вид $F = \exists x_1 X$. В соответствии с описанной процедурой преобразованная формула будет иметь вид $\tilde{F} = X_{c_1}^{x_1}$. Ложность F равносильна истинности $\neg F \equiv \forall x_1 \neg X$ или, что то же самое, истинности $\neg X$ при любой интерпретации и любой оценке. Замена x_1 на константу лишь переносит конкретизацию значения x_1 из сферы оценки в сферу интерпретации.

Предположим, что формула F имеет вид $F = \forall x_1 \forall x_2 \dots \forall x_{r-1} \exists x_r X$. В соответствии с правилом удаления квантора существования преобразованная формула будет иметь вид $\tilde{F} = \forall x_1 \forall x_2 \dots \forall x_{r-1} X_{f(x_1, x_2, \dots, x_{r-1})}^{x_r}$. Ложность формулы F в данной интерпретации означает, что существует такая оценка $\vartheta = \begin{pmatrix} x_1 & x_2 & \dots & x_{r-1} \\ c_1 & c_2 & \dots & c_{r-1} \end{pmatrix}$, что формула $\exists x_r X \vartheta$ ложна. Значит, формула $X \vartheta$ ложна при любой оценке переменной x_r . Но тогда независимо от интерпретации символа f формула $X \vartheta_{f(c_1 \dots c_{r-1})}^{x_r}$, или $(X_{f(x_1 \dots x_{r-1})}^{x_r}) \vartheta$ ложна. Значит, формула $\tilde{F}$ ложна в данной интерпретации.

Пусть в некоторой интерпретации формула F истинна. Значит, для любой оценки $\vartheta = \begin{pmatrix} x_1 & x_2 & \dots & x_{r-1} \\ c_1 & c_2 & \dots & c_{r-1} \end{pmatrix}$ формула $\exists x_r X \vartheta$ истинна, т.е. каждой оценке ϑ соответствует значение $x_r(\vartheta)$, при котором формула X истинна. Следовательно, определена функция $x_r(x_1, x_2, \dots, x_{r-1})$, которая при $x_1 = c_1, \dots, x_{r-1} = c_{r-1}$ принимает значение, равное соответствующему значению x_r . Добавив к заданной интерпретации интерпретацию символа f как указанной функции, получим интерпретацию, в которой формула $\tilde{F}$ будет истинной. ►

Замечание 7.1. Не следует считать, что скунемовская стандартная форма данной формулы эквивалентна этой формуле. Речь идет лишь о том, что тождественная ложность (т.е. в любой интерпретации) исходной формулы равносильна тождественной ложности ее скунемовской стандартной формы. Рассмотрим простейший пример $F = \exists x p(x)$. Скунемовской стандартной формой будет $F_s = p(a)$. Выбрав носитель интерпретации $\{1, 2\}$ и положив $|p(1)| = 0, |p(2)| = 1$, заключаем, что формула F будет истинной. Выбрав в качестве интерпретации константы a элемент 1, приходим к выводу, что F_s будет ложной. Значит $F \neq F_s$.

Скунемовская форма не сохраняет даже свойство формулы быть тавтологией. Например, формула $\exists x_1 \exists x_2 (p(x_1) \vee \neg p(x_2))$ является тавтологией, в то время как ее скунемовская форма $(p(c_1) \vee \neg p(c_2))$ при определенной интерпретации предикатного символа p и констант c_1, c_2 оказывается ложной. #

7.2. Метод резолюций для исчисления высказываний

Ставится задача построения процедуры, которая за конечное число шагов даст доказательство истинности конкретной формулы. Подход, основанный на построении таблиц истинности, который работает в исчислении высказываний, в исчислении предикатов неприемлем. Но доказательство того, что любая непротиворечивая теория интерпретируема, дает идею для автоматического поиска доказательства. Отметим, что речь не идет о построении вывода в той или иной формальной системе, а лишь о проверке того, что данная формула есть тавтология. Точнее, речь идет не о проверке, а построении некоей процедуры, которая подтвердит, что данная формула есть тавтология, если это действительно так.

Оказывается такую процедуру формирует синтаксический разбор, проведенный в доказательстве теоремы о непротиворечивой теории. Отметим, что задача доказать тавтологичность формулы равносильна задаче доказать, что отрицание данной формулы является противоречием. А проверку формулы на противоречивость можно провести, используя ее скунемовскую стандартную форму, в которой не используется импликация, а отрицания относятся непосредственно к атомам формулы.

Подобный подход привел к разработке метода, известного как **метод резолюций**. Использование этого метода начнем с алгебры высказываний.

Рассмотрим некоторое множество Γ дизъюнктов. Мы можем считать, что в каждом дизъюнкте литера с данной переменной x входит только один раз, поскольку дизъюнкция одинаковых литер есть сама эта литера, а дизъюнкция **контрарных литер** (т.е. одна является отрицанием другой) дает дизъюнкт, являющийся тавтологией. Такой дизъюнкт не влияет на противоречивость всего множества и его можно из рассмотрения убрать.

Предположим, что $\Gamma = \{D_1, D_2, \dots, D_k\}$, причем дизъюнкт D_1 содержит переменную p , а дизъюнкт D_2 — литеру $\neg p$, т.е. $D_1 = p \vee \tilde{D}_1$, $D_2 = \neg p \vee \tilde{D}_2$, где $\tilde{D}_1$ и $\tilde{D}_2$ не содержат переменной p . Такие дизъюнкты будем называть **контрарной парой**. Тогда можно рассмотреть дизъюнкт $\tilde{D}_1 \vee \tilde{D}_2$ вообще не содержащий переменной p . Этот дизъюнкт называется **резольвентой** дизъюнктов D_1, D_2 .

Введем очевидное понятие: формула B называется логическим следствием формулы A , если формула $A \rightarrow B$ является тавтологией. В исчислении высказываний (и в исчислении предикатов тоже) это равносильно существованию выводимости $A \vdash B$ или выводимости секвенции $A \vdash B$. Множество дизъюнктов Δ назовем логическим следствием множества дизъюнктов Γ , если конъюнкция всех дизъюнктов Δ есть логическое следствие конъюнкции всех дизъюнктов Γ .

Теорема 7.2. Пусть дизъюнкты D_1 и D_2 образуют контрарную пару. Тогда их резольвента является логическим следствием множества дизъюнктов $\{D_1, D_2\}$.

◀ Пусть $D_1 = p \vee \tilde{D}_1$, $D_2 = \neg p \vee \tilde{D}_2$. Нам нужно убедиться в истинности формулы $D_1 \wedge D_2 \rightarrow (\tilde{D}_1 \vee \tilde{D}_2)$, что равносильно выводимости генценовской секвенции $D_1, D_2 \vdash \tilde{D}_1, \tilde{D}_2$. Подставим в эту секвенцию представление дизъюнктов D_1 и D_2 : $p \vee \tilde{D}_1, \neg p \vee \tilde{D}_2 \vdash \tilde{D}_1, \tilde{D}_2$. Имеем:

$$\begin{array}{c}
 p \vee \tilde{D}_1, \neg p \vee \tilde{D}_2 \vdash \tilde{D}_1, \tilde{D}_2 \\
 \swarrow \quad \searrow \\
 p, \neg p \vee \tilde{D}_2 \vdash \tilde{D}_1, \tilde{D}_2 \quad \tilde{D}_1, \neg p \vee \tilde{D}_2 \vdash \tilde{D}_1, \tilde{D}_2 \\
 \swarrow \quad \searrow \quad \quad \swarrow \quad \searrow \\
 p, \neg p \vdash \tilde{D}_1, \tilde{D}_2 \quad p, \tilde{D}_2 \vdash \tilde{D}_1, \tilde{D}_2 \\
 \downarrow \\
 p, \vdash \tilde{D}_1, \tilde{D}_2, p
 \end{array}$$

Итак, секвенция $p \vee \tilde{D}_1, \neg p \vee \tilde{D}_2 \vdash \tilde{D}_1, \tilde{D}_2$ выводима в генценовском исчислении. Следовательно, множество из одного дизъюнкта $\{\tilde{D}_1 \vee \tilde{D}_2\}$ есть логическое следствие множества дизъюнктов $\{D_1, D_2\}$. ▶

Доказанная теорема приводит к следующей процедуре обоснования противоречивости множества дизъюнктов. Отталкиваясь от исходного множества S_0 , находим контрарную пару, конструируем ее резольвенту и добавляем к S_0 , получая S_1 . С множеством S_1 поступаем аналогично, пропуская, правда, уже рассмотренную контрарную пару. Для автоматизации процесса можно предложить следующий алгоритм. Множество рассматриваем как упорядоченный список, резольвенты добавляем в конец списка, а текущее состояние всего списка фиксируем парой индексов — номер рассматриваемой пары в списке. Завершением этой процедуры является получение пустого дизъюнкта, что и является доказательством противоречивости исходного множества.

Действительно, каждая вновь получаемая резольвента есть логическое следствие текущего множества S_i дизъюнктов. Значит, ее добавление к множеству дает новое множество, которое с точки зрения противоречивости эквивалентно исходному. Если на очередном шаге возникает пустой дизъюнкт, то последнее множество дизъюнктов противоречиво, а значит, противоречиво и исходное множество дизъюнктов. В этом и состоит суть **метода резолюций**.

Поскольку возникающий пустой дизъюнкт есть резольвента двух уже имеющихся дизъюнктов, которые в свою очередь есть либо дизъюнкты исходного множества, либо резольвенты. Возникает дерево, отражающее процесс построения пустого дизъюнкта из имеющегося множества дизъюнктов. Это приводит к следующему определению.

Последовательность дизъюнктов, в которой каждый дизъюнкт либо принадлежит множеству Γ , либо является резольвентой двух предыдущих дизъюнктов, называется **резольвентивным выводом из множества Γ** . Конечный дизъюнкт резольвентивного вывода из множества Γ называется **выводимым из множества Γ** .

Пример 7.3. Рассмотрим множество $\Gamma = \{\neg x \vee y, \neg y, x\}$. Тогда резолютивным выводом для этого множества является следующий:

- $\neg x \vee y$ — дизъюнкт множества Γ ;
- x — дизъюнкт множества Γ ;
- y — резольвента двух предыдущих;
- $\neg y$ — дизъюнкт множества Γ ;
- $\square$ — резольвента двух предыдущих.

Отметим, что получение пустого дизъюнкта как результата вывода означает, что исходное множество дизъюнктов Γ противоречиво. Действительно, последовательно добавляя к Γ формулы из вывода, получаем последовательность расширяющихся множеств дизъюнктов, в которой каждое очередное множество есть логическое следствие предыдущего. Последнее множество содержит пустой дизъюнкт и, следовательно, является противоречивым. Поэтому и исходное множество дизъюнктов является противоречивым.

Теорема 7.3. Множество Γ дизъюнктов противоречиво тогда и только тогда, когда существует вывод из этого множества пустого дизъюнкта.

◀ В примере фактически уже было доказано, что если существует вывод пустого дизъюнкта, то исходное множество дизъюнктов противоречиво. Чтобы доказать противоположное утверждение, рассмотрим некоторое противоречивое множество дизъюнктов Γ . Добавляя к нему последовательно резольвенты различных контрарных пар, получаем расширение исходного множества с сохранением его противоречивости. Это расширение потихоньку поглощает дерево вывода пустого дизъюнкта, существование которого необходимо доказать. Что может быть мерой близости к пустому дизъюнкту? Например, количество литер в очередной резольвенте. Чем короче очередной получаемый дизъюнкт, тем ближе дизъюнкт нулевой доины, т.е. пустой.

Процедуру перехода ко все более коротким дизъюнктам можно описать геометрически. Все переменные, встречающиеся в исходном списке дизъюнктов Γ , расположим в виде последовательности: $x_1, x_2, \dots, x_k$. Перебор всех вариантов значений этих переменных можно представить в виде дерева, в котором метка дуги x_i означает, что $x_i = 1$ (истинно), а метка $\neg x_i$ — что $x_i = 0$ (ложно) (рис. 7.1).

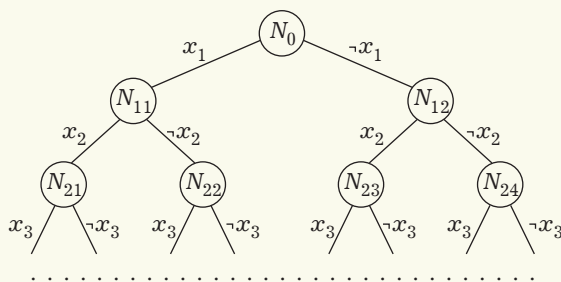


Рис. 7.1

Листьям этого бинарного дерева соответствуют конкретные наборы значений переменных. По условию при каждом таком наборе значений один из дизъюнктов противоречив, поскольку конъюнкция всех дизъюнктов является противоречием. Рассмотрим пару листьев, выходящих из одного узла: L_1 с $x_n = 0$ и L_2 с $x_n = 1$. Пусть им соответствуют дизъюнкты D_1 и D_2 . Возможны два варианта. В первом варианте один из дизъюнктов, например D_1 , противоречив для обоих листьев. Тогда D_1 не содержит переменной x_n . Действительно, если $D_1 = \tilde{D}_1 \vee x_n^0$, то при $x_n = \sigma$, т.е. для одного из двух листов, дизъюнкт D_1 будет истинным. В этом варианте удаляем два листа, а дизъюнкт D_1 приписываем узлу, порождающему листья L_1 и L_2 . После удаления листьев, порождающих их узел станет листом.

Во втором варианте дизъюнкт D_1 противоречив в листе L_1 (т.е. при $x_n = 0$) и истинен в листе L_2 , а дизъюнкт D_2 противоречив в листе L_2 и истинен в L_1 . Тогда D_1 имеет вид

$D_1 = \tilde{D}_1 \vee x_n$, а D_2 имеет вид $D_2 = \tilde{D}_2 \vee \neg x_n$. Резольвента D двух дизъюнктов, имеющая вид $D = \tilde{D}_1 \vee \tilde{D}_2$, противоречива и в L_1 , и в L_2 . Удаляем листья L_1, L_2 , а порождающему их узлу ставим в соответствие резольвенту.

Продолжая процедуру отрезания пар листьев, мы последовательно сокращаем размер дерева, причем на каждом шаге каждому листу соответствует дизъюнкт, являющийся противоречивым при соответствующих значениях переменных. В этот дизъюнкт входят только те переменные, которые однозначно определяются узлом исходного дерева. Отрезав все листья, получим дерево из одного корня, которому соответствует пустой дизъюнкт. ►

7.3. Метод резолюций для исчисления предикатов

В случае исчисления предикатов резолютивный вывод касается бескванторных формул, но под литерой дизъюнкта надо понимать элементарную формулу, в которую в качестве аргументов предикатного символа могут входить различные термы. Это усложняет ситуацию, поскольку один предикатный символ в зависимости от комбинации термов в его аргументах может давать разные истинностные значения. Но правило резолюций остается тем же: если есть дизъюнкты $P \vee D_1$ и $\neg P \vee D_2$, то их заменяем резольвентой $D_1 \vee D_2$. Здесь P — некоторая элементарная формула.

Однако только такого склеивания дизъюнктов недостаточно. Отметим, что нашей целью является установление противоречивости множества дизъюнктов. С этой целью мы довольно свободно можем менять переменные в аргументах предикатных символов данной формулы. Реализуется это механизмом подстановки. Идея такова: подобрать такую подстановку, что элементарные формулы, содержащие данный предикатный символ, преобразовались в одну и ту же формулу. Подстановка не нарушает условия противоречивости (или тавтологичности).

Пусть дано множество формул $\Phi_1, \Phi_2, \dots, \Phi_k$. Подстановка ϑ называется **унификатором** этого множества, если $\Phi_1\vartheta = \Phi_2\vartheta = \dots = \Phi_k\vartheta$. Если множество имеет унификатор, то оно называется унифицируемым.

Пример 7.4. Множество $\{p(a, y), p(x, f(b))\}$ имеет унификатор $\begin{pmatrix} x & y \\ a & f(b) \end{pmatrix}$. Следовательно, оно унифицируемо.

Пример 7.5. Множество из формул $p(a, y); p(b, f(x))$ не имеет унификаторов, поскольку любая подстановка меняет переменные, но не константы. Следовательно, при любой подстановке ϑ первая формула будет иметь вид $p(a, T_1)$, а вторая $p(b, T_2)$, где T_1 и T_2 — некоторые термы соответствующего сорта. Значит, никакая подстановка не приведет к тому, что формулы станут одинаковыми.

Подстановки можно применять последовательно. Пусть есть подстановки $\vartheta_1 = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ t_1 & t_2 & \dots & t_n \end{pmatrix}$ и $\vartheta_2 = \begin{pmatrix} y_1 & y_2 & \dots & y_n \\ q_1 & q_2 & \dots & q_n \end{pmatrix}$. Их композицией $\vartheta_1 \circ \vartheta_2$ называется подстановка

$$\vartheta_1 \circ \vartheta_2 = \begin{pmatrix} x_1 & x_2 & \dots & x_n & y_1 & y_2 & \dots & y_n \\ t_1\vartheta_2 & t_2\vartheta_2 & \dots & t_n\vartheta_2 & q_1 & q_2 & \dots & q_n \end{pmatrix}$$

(запись $t_i\vartheta_2$ означает, что в терме t_i надо выполнить подстановку ϑ_2). Ясно, что композиция подстановок реализует последовательное применение двух подстановок к формуле, причем последовательность применения — слева направо.

Унификатор ϑ множества формул $\Phi_1, \Phi_2, \dots, \Phi_k$ называется **наиболее общим**, если любой унификатор λ этого множества можно представить в виде $\lambda = \vartheta \circ \sigma$ с помощью некоторой подстановки σ .

Унификация — процедура, позволяющая (при некоторых условиях) сохранять свойство противоречивости, но при этом согласовывать аргументы у одинаковых предикатных символов.

Алгоритм, позволяющий для любого конечного множества элементарных формул с одинаковым предикатным символом построить наиболее общий унификатор или доказать, что это множество не унифицируемо, довольно прост. Его идея заключается в том, чтобы найти рассогласования в формулах, а потом найти подстановку, которая это рассогласование устраняет (ну, или убедиться в том, что это рассогласование неустранимо). Например, в формулах $p(x, b)$ и $p(a, b)$ рассогласование заключается в том, что на местах, в которых в первой формуле находится переменная x , вторая формула содержит константу a . Рассогласование можно устранить подстановкой вместо x константы a .

Рассматриваем множество элементарных формул вида $p(T_1, T_2, \dots, T_n)$ с одинаковым предикатным символом. У таких формул последовательности аргументов имеют одинаковую длину и одинаковые последовательности сортов. Мы сравниваем у всех формул первые аргументы, затем вторые и т.д., пока не найдем очередную серию аргументов, в которой есть неодинаковые термы. Эти термы могут порождаться одним и тем же функциональным символом. Тогда мы переходим к сравнению аргументов этого функционального символа. Происходит анализ всего синтаксического дерева. Формально это можно провести так. Просматриваем формулы как последовательности символов слева направо до появления на очередной позиции разных символов. В каждой формуле с этой позиции начинается некоторый подтерм. Список таких подтермов и есть список рассогласования.

Пример 7.6. В множестве формул $p(x, f(y, z)), p(x, a), p(x, g(h(k(x))))$ различие начинается с позиции 5. С этой позиции в первой формуле начинается терм $f(y, z)$, во второй — терм a , в третьей — терм $g(h(k(x)))$. Рассогласование в данном случае — список термов $f(y, z)$, a , $g(h(k(x)))$.

Рассмотрим множество формул $p(x, f(y, z)), p(x, f(g(x), z)), p(x, f(a, x))$. В данном случае формулы имеют одинаковый первый аргумент, а второй у них различается. Но при этом у них у всех этот аргумент порожден одним и тем же функциональным символом f . Различие пор символом наступает в позиции 7, в которой начинается первый аргумент символа f . Выписывая по всем формулам эти аргументы, получаем рассогласование $y, g(x), a$.

Рассмотрим некоторое множество элементарных формул $p(T_1^j, T_2^j, \dots, T_k^j)$, $j = 1, 2, \dots, s$. Находим список рассогласования и в нем определяем два терма, один из которых есть переменная x , а второй есть терм T . Во всех формулах выполняем подстановку $\left(\begin{smallmatrix} x \\ T \end{smallmatrix}\right)$. Процедуру повторяем. Если в какой-то момент очередной список рассогласования не имеет переменных, то множество формул не унифицируемо. Другой вариант завершения процедуры — пустой список рассогласования (формулы стали одинаковыми). Тогда множество унифицируемо, а наиболее общий унификатор — композиция всех выполненных подстановок.

Пример 7.7. 1. Рассмотрим множество из двух формул $p(a, x, f(g(y)))$, $p(z, f(z), f(u))$. Первый список рассогласований: $\{a, z\}$. Он устраняется подстановкой $\left(\begin{smallmatrix} z \\ a \end{smallmatrix}\right)$. В результате получим множество $p(a, x, f(g(y)))$, $p(a, f(a), f(u))$. Следующий список рассогласований $\{x, f(a)\}$ устраняется подстановкой $\left(\begin{smallmatrix} x \\ f(a) \end{smallmatrix}\right)$. Реализация этой подстановки дает множество $p(a, f(a), f(g(y)))$, $p(a, f(a), f(u))$. Следующий список рассогласований $\{g(y), u\}$. Устраняющая подстановка $\left(\begin{smallmatrix} u \\ g(y) \end{smallmatrix}\right)$. Реализуем ее: $p(a, f(a), f(g(y)))$, $p(a, f(a), f(g(y)))$. Получили одинаковые формулы. процесс унификации закончен. Унификатор: $\left(\begin{smallmatrix} z & x & u \\ a & f(a) & g(y) \end{smallmatrix}\right)$.

2. Рассмотрим множество $q(f(a), g(x)), q(y, y)$. Первый список рассогласования $\{f(a), y\}$ устраняется подстановкой $\left(\begin{smallmatrix} y \\ f(a) \end{smallmatrix}\right)$. В результате реализации этой подстановки получим множество $q(f(a), g(x)), q(f(a), f(a))$. Второй список рассогласования $\{g(x), f(a)\}$ не может быть устранен, поскольку среди термов нет ни одной переменной. Вывод: рассматриваемое множество не унифицируемо.

Пусть в дизъюнкте C несколько литер $L_1, L_2, \dots, L_k$ с предикатным символом p одной полярности (т.е. либо все с отрицанием, либо все без отрицания). Предположим, что они имеют наиболее общий унификатор ϑ . Тогда формулы $L_1\vartheta, L_2\vartheta, \dots, L_k\vartheta$ совпадают и в дизъюнкте $C\vartheta$ заменяются одной литерой. Дизъюнкт $C\vartheta$ в этом случае называется склейкой дизъюнкта C .

Пусть C_1 и C_2 — два дизъюнкта, не имеющих одинаковых переменных. Предположим, что в них входят литеры вида $p(\dots)$ и $\neg p(\dots)$, для которых существует наиболее общий унификатор ϑ , т.е., например, $C_1 = p(\dots) \vee \tilde{C}_1$, $C_2 = \neg p(\dots) \vee \tilde{C}_2$. После подстановки заменим литеры p совпадут, и мы можем сформировать резольвенту $C_1\vartheta \vee C_2\vartheta$, которая называется **бинарной резольвентой** C_1 и C_2 .

Мы рассматриваем множество дизъюнктов как „скелет“ матрицы скулемовской стандартной формы. Значит, в действительности все переменные связаны кванторами всеобщности. Но отметим, что

$$\forall x(D_1(x) \wedge D_2(x)) \equiv \forall x D_1(x) \wedge \forall x D_2(x) \equiv \forall x_1 D_1(x_1) \wedge \forall x_2 D_2(x_2) \equiv \forall x_1 \forall x_2 (D_1(x_1) \wedge D_2(x_2)).$$

Это показывает, что скулемовскую стандартную форму можно составить так, что во всех ее дизъюнктах будут разные переменные.

Резольвентой двух дизъюнктов C_1, C_2 называется бинарная резольвента C_1 или его склейки и C_2 или его склейки. Здесь четыре ситуации: дизъюнкт–дизъюнкт, склейка–дизъюнкт, дизъюнкт–склейка и склейка–склейка.

Пример 7.8. В множестве формул $\neg p(x, y), p(a, b) \vee p(a, z) \vee p(f(x), f(x))$ во втором дизъюнкте возможна склейка второй литеры с первой, а третья литера не сливается ни с чем. После склейки получим формулу $p(a, b) \vee p(f(x), f(x))$, которая имеет резольвенту с первой формулой $p(f(a), f(a))$.

Пример 7.9. Рассмотрим формулы $F_1 = \forall x(c(x) \rightarrow w(x) \wedge r(x))$; $F_2 = \exists x(c(x) \wedge q(x))$; $G = \exists x(q(x) \wedge r(x))$. Покажем, что G есть логическое следствие F_1 и F_2 . Это значит, что множество формул $F_1, F_2, \neg G$ противоречиво.

Преобразуем формулы рассматриваемого множества в скулемовскую стандартную форму:

$$\begin{aligned} F_1 &\simeq \forall x(\neg c(x) \vee (w(x) \wedge r(x))) \simeq \forall x([\neg c(x) \vee w(x)] \wedge [\neg c(x) \vee r(x)]); \\ F_2 &\simeq c(a) \wedge q(a); \\ \neg G &\simeq \forall x(\neg q(x) \vee \neg r(x)). \end{aligned}$$

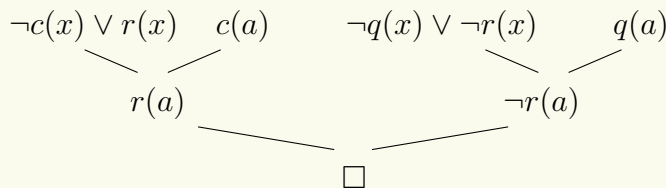
Требуется опровержение формулы $F_1 \wedge F_2 \wedge \neg G$, которая имеет скулемовскую форму

$$F_1 \wedge F_2 \wedge \neg G \simeq \forall x([\neg c(x) \vee w(x)] \wedge [\neg c(x) \vee r(x)] \wedge c(a) \wedge q(a) \wedge [\neg q(x) \vee \neg r(x)]).$$

Отбрасывая квантор, получим множество дизъюнктов

$$\neg c(x) \vee w(x), \quad \neg c(x) \vee r(x), \quad c(a), \quad q(a), \quad \neg q(x) \vee \neg r(x).$$

Для доказательства противоречивости этого множества используем метод резолюций:



Поскольку методом резолюций получен вывод пустого дизъюнкта, заключаем, что множество дизъюнктов противоречиво, а следовательно, формула G есть логическое следствие F_1 и F_2 .

Пример 7.10. Рассмотрим формулы

$$F_1 = \exists x(p(x) \wedge \forall y[d(y) \rightarrow r(x, y)]), \quad F_2 = \forall x(p(x) \rightarrow \forall y[q(y) \rightarrow \neg r(x, y)]), \quad G = \forall x[d(x) \rightarrow \neg q(x)].$$

Выясним, является ли формула G логическим следствием F_1 и F_2 .

Приведем формулы к скунемовской стандартной форме. Для формулы F_1 имеем:

$$F_1 \equiv \exists x \forall y(p(x) \wedge [d(y) \rightarrow r(x, y)]) \equiv \exists x \forall y(p(x) \wedge [\neg d(y) \vee r(x, y)]) \simeq \forall y(p(a) \wedge [\neg d(y) \vee r(a, y)]).$$

Для формулы F_2 получаем

$$F_2 \equiv \forall x \forall y(p(x) \rightarrow [q(y) \rightarrow \neg r(x, y)]) \equiv \forall x \forall y(\neg p(x) \vee \neg q(y) \vee \neg r(x, y)).$$

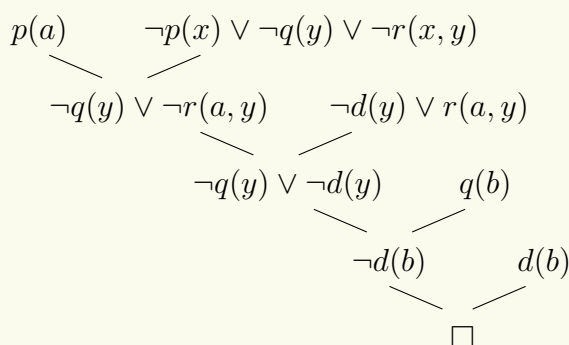
Наконец, преобразуем отрицание формулы G :

$$\neg G \equiv \exists x \neg[d(x) \rightarrow \neg q(x)] \equiv \exists x \neg[\neg d(x) \vee \neg q(x)] \equiv \exists x[d(x) \wedge q(x)] \simeq d(b) \wedge q(b).$$

В результате получаем следующее множество дизъюнктов:

$$p(a), \quad \neg d(y) \vee r(a, y), \quad \neg p(x) \vee \neg q(y) \vee \neg r(x, y), \quad d(b), \quad q(b).$$

Приведем дерево резольвентного вывода:



В результате множество дизъюнктов противоречиво, т.е. формула $F_1 \wedge F_2 \rightarrow G$ является тавтологией, а формула G есть логическое следствие F_1 и F_2 .

ОГЛАВЛЕНИЕ

1. Алгебра высказываний	2
1.1. Введение	2
1.2. Алгебра логики	4
1.3. Тавтологии и эквивалентность формул	6
1.4. Функции алгебры логики	10
2. Исчисление высказываний	15
2.1. Основные положения теории N	15
2.2. Правила естественного вывода	17
2.3. Глобальные свойства теории N	22
3. Алгебра предикатов	27
3.1. Предикаты и кванторы	27
3.2. Логико-математические языки	28
3.3. Переименования и подстановки	31
3.4. Семантика логико-математического языка	34
3.5. Логические законы	36
3.6. Замены	39
3.7. Упрощение формул	41
4. Исчисление предикатов	43
4.1. Построение теории P	43
4.2. Правила естественного вывода	44
4.3. Глобальные свойства теории P	46
5. Генценовские формальные системы	54
5.1. Исчисление GV	54
5.2. Исчисление GP	58
6. Примеры формальных теорий	61
6.1. Теория групп	61
6.2. Формальная арифметика	64
7. Метод резолюций	70
7.1. Скулемовские функции	70
7.2. Метод резолюций для исчисления высказываний	72
7.3. Метод резолюций для исчисления предикатов	75