Московский физико-технический институт
Физтех-школа прикладной математики и информатики
Математическая логика и теория алгоритмов, осень 2021
Лекция 1: формальный язык записи логических утверждений

Краткое содержание

Необходимость в формальной системе записи математических утверждений. Основные понятия теории формальных языков: алфавит, слово, язык. Пустое слово. Операции со словами: конкатенация, обращение. Отношения на словах: префиксы, суффиксы, подслова, подпоследовательности. Операции с языками: теоретико-множественные, конкатенация, звезда и плюс Клини. Три определения правильной скобочной последовательности: через скобочный итог, через разбиение скобок на пары и рекурсивное. Их эквивалентность. Неоднозначность разбора при рекурсивном определении. Алфавит алгебраических выражений: переменные, символы операций, скобки. Рекурсивное построение правильных алгебраических выражений. Лемма о скобочном итоге и теорема об однозначности разбора для правильных алгебраических выражений. Алфавит логики высказываний: пропозициональные переменные, символы логических операций, скобки. Правила построения пропозициональных формул. Лемма о скобочном итоге и теорема об однозначности разбора для пропозициональных формул.

1 Зачем нужен формальный язык

По мере развития математики её язык совершенствовался. Постепенно появились различные обозначения: буквы в качестве переменных, символы операций, стандартные названия функций и т.д. Однако собственно математические (в частности, логические) рассуждения долго проводились на естественном языке. К концу XIX – началу XX вв. возникла потребность в формализации рассуждений: с одной стороны, формальная система записи облегчала коммуникацию между разными математиками, с другой стороны, были открыты парадоксы, связанные с рассуждениями на естественном языке. Представлялось, что аксиоматизация математики и полная формализация рассуждений поможет от них избавиться.[1]

Примером такого парадокса может служить парадокс Рассела. Назовём *правильным* множество, которое не является собственным элементом: например, множество натуральных чисел само не является натуральным числом, а множество всех треугольников не является треугольником. С другой стороны, есть и *неправильные* множества. Например, множество всех множеств является множеством, а множество всех бесконечных множеств является бесконечным множеством. Вопрос: каким является множество правильных множеств? Если оно правильное, то оно входит в множество правильных множеств, т.е. является собственным элементом, т.е. является неправильным. Если же оно неправильное, то является собственным элементом, т.е. входит в множество правильных множеств, т.е. является правильным. Таким образом, в любом случае получается противоречие. Впрочем, этот парадокс разрешается не формализацией рассуждений, а аксиоматизацией теории множеств, ограничивающей способы формирования

[1] О сведе́нии любого рассуждения к механической процедуре, вычислению говорил ешё Лейбниц, но даже с самой постановкой вопроса он значительно опередил своё время.

множеств. Зато без формализации нельзя даже сформулировать, например, вторую проблему Гильберта: являются ли аксиомы арифметики непротиворечивыми?

Формальные системы записи математических утверждений и рассуждений были в целом построены в первой половине XX в. и сыграли очень большую роль в последовавшем развитии вычислительной техники. Во-первых, компьютеры не могут понимать никаких языков, кроме формальных, поэтому знание теории формальных языков очень помогает при программировании и, особенно, написании компиляторов. Во-вторых, формальная запись математических утверждений и доказательств позволяет искать и проверять доказательства машинными методами. Исторически первым и, пожалуй, самым известным примером такого доказательства является решение проблемы четырёх красок: верно ли, что любую географическую карту можно покрасить в 4 цвета, так чтобы области одного цвета не граничили друг с другом? Первоначально проблема была положительно решена в 1976 году Аппелем и Хакеном путём компьютерного перебора. Впоследствии доказательство было несколько раз упрощено, хотя и оставалось переборным, а в 2005 году Вернер и Гонтье записали полное доказательство на формальном языке и верифицировали его при помощи программы Coq. Сегодня верификация на Coq стала стандартным инструментом и в математике, и в анализе программ.

В этой лекции мы коснёмся самых основ теории формальных языков, более подробно она будет освещена в курсе «Формальные языки и трансляции».

2 Основные понятия теории формальных языков

2.1 Символ→Алфавит→Слово→Язык

Определение 1. *Алфавитом* называется любое конечное непустое множество. Элементы алфавита называются *символами*.

Как правило, алфавиты мы будем обозначать большими греческими буквами: Σ, Γ и т.д., а символы алфавита — маленькими греческими: σ, τ и т.д., — или латинскими буквами из начала алфавита: a, b, c, — или конкретными значками, например, скобками. Алфавит $\{0, 1\}$ будем называть *бинарным* или *двоичным*. Теоретически можно рассмотреть и бесконечный алфавит (и даже несчётный), но по умолчанию мы этого делать не будем.

Определение 2. *Словом* называется любая конечная последовательность (цепочка, кортеж) символов. Число символов в этой последовательности называется *длиной* слова.

При записи слов, в отличие от последовательностей, мы не будем использовать скобки и запятые, просто записывая символы один за другим: $\sigma_1\sigma_2\ldots\sigma_n$. Как правило, слова мы будем обозначать маленькими латинскими буквами из конца алфавита: u, v, x и т.д. Длину слова u (т.е. число символов) мы будем обозначать через $|u|$ (также используют обозначения $l(u)$ или $\mathrm{len}(u)$). Особое значение играет *пустое слово*, т.е. слово нулевой длины, последовательность нуля символов. Оно ровно одно и не зависит от алфавита,

поэтому мы используем для него специальное обозначение ε (иногда также пишут λ или Λ).

Два слова называются равными, если они состоят из одних и тех же символов в одинаковом порядке. Иначе говоря, $u = v$, если $|u| = |v|$ и i-е символы u и v совпадают для любого i.[2] В некоторых контекстах для обозначения такого равенства используется имвол $\eqcirc$. Такое равенство называется графическим, или посимвольным, его не следует путать с арифметическим или алгебраическим равенством. Например, $2 + 2$ равно 4 в арифметическом смысле, но не посимвольно. А выражения $(x+y)(x-y)$ и x^2-y^2 равны в алгебраическом смысле, но вновь не посимвольно.

Определение 3. *Языком* называется любое множество слов в некотором алфавите.

Языки могут быть и конечными, и бесконечными. Языки мы будем обозначать большими латинскими буквами: L, M и т.д. Язык может быть пустым, в таком случае его обозначают $\varnothing$, как пустое множество (иногда используют обозначение Φ). Не следует путать пустой язык и язык из пустого слова, т.е. $\{\varepsilon\}$ (последний ещё называют *синглетоном*).

2.2 Операции над словами

Над словами можно проводить некоторые операции, базовой является конкатенация.

Определение 4. Пусть $u = \sigma_1 \dots \sigma_k$ и $v = \tau_1 \dots \tau_l$ суть два слова. Тогда *конкатенацией* слов u и v называется слово $u{\cdot}v = \sigma_1 \dots \sigma_k \tau_1 \dots \tau_l$. Иными словами, если $|u| = k$, а $|v| = l$, то $|u \cdot v| = k + l$, а i-й символ $u \cdot v$ равняется i-му символу u при $i \leqslant k$ и $(i - k)$-му символу v при $i > k$.

Суть определения заключается в том, что слово v приписывается справа к слову u. Также можно отметить, что конкатенация очень похожа на умножение. Если каждый символ рассматривать как переменную, а слова — как произведения переменных, то перемножаться они должны именно так. Поэтому знак $\cdot$ для конкатенации, как и для умножения, часто опускают. Подобно умножению, конкатенация порождает понятие степени.

Определение 5. Пусть u — некоторое слово, а n — натуральное[3] число. Тогда его n-й степенью называется слово $u^n = \underbrace{u \cdot \dots \cdot u}_{n \text{ раз}}$.

[2] Именно по этому определению любые два пустых слова равны: их длины равны нулю, а символы на любых позициях совпадают. Почему так? Тут мы встречаемся с ещё одним важным логическим принципом: элемент пустого множества обладает любыми свойствами. Невозможно указать позицию, на которой слова различаются, значит, они на всех совпадают.

[3] Мы предполагаем, что натуральные числа начинаются с нуля, т.е. $\mathbb{N} = \{0, 1, 2, 3, \dots\}$. Это соответствует пониманию, что натуральное число отвечает на вопрос «Сколько?» и обозначает количество. В средней школе обычно говорят, что натуральные числа начинаются с единицы: это значит, что число отвечает на вопрос «Какой по счёту?» и обозначает порядковый номер. Выбор одного из двух подходов является делом вкуса, но в логике и информатике удобнее первый.

При этом нулевая степень любого слова, т.е. конкатенация нуля слов, по определению считается пустым словом. Можно также определять степень итеративно: $u^0 = \varepsilon$, а при $n > 0$ выполнено $u^n = u^{n-1} \cdot u$.

Отметим следующие простые факты:

Утверждение 6. *Для всех слов u, v и w выполнены следующие свойства:*

а) $(u \cdot v) \cdot w = u \cdot (v \cdot w)$ *(ассоциативность конкатенации);*

б) $u \cdot \varepsilon = \varepsilon \cdot u = u$ *(пустое слово является нейтральным элементом);*

в) $u^n \cdot u^m = u^{n+m}$*;*

г) $(u^n)^m = u^{nm}$*;*

д) $\varepsilon^n = \varepsilon$.

В алгебраических терминах можно сказать, что слова образуют моноид, или полугруппу с нейтральным элементом относительно конкатенации. (И именно поэтому она похожа на умножение). Более того, эта полугруппа свободная.

Доказательство. Для примера докажем первое утверждение. Пусть $|u| = k$, $|v| = l$, $|w| = m$. Тогда $|(u \cdot v) \cdot w| = (k + l) + m$, а $|u \cdot (v \cdot w)| = k + (l + m)$. Значит, длины в обоих вариантах совпадают. Далее, пусть $i \leqslant k$. Тогда $i \leqslant k + l$, поэтому i-й символ $(u \cdot v) \cdot w$ равен i-му символу $(u \cdot v)$ и потому равен i-му символу u. С другой стороны, i-й символ $u \cdot (v \cdot w)$ тоже равняется i-му символу u. Пусть теперь $k < i \leqslant k + l$. Тогда i-й символ $(u \cdot v) \cdot w$ по-прежнему равен i-му символу $(u \cdot v)$, но теперь равен $(i - k)$-му символу v. С другой стороны, i-й символ $u \cdot (v \cdot w)$ равен $(i - k)$-му символу $v \cdot w$ и в силу $i - k \leqslant l$ равен $(i - k)$-му символу v. Наконец, пусть $i > k + l$. Тогда i-й символ $(u \cdot v) \cdot w$ равен $(i - (k + l))$-му символу w. С другой стороны, т.к. $i > k$, i-й символ $u \cdot (v \cdot w)$ равен $(i - k)$-му символу $v \cdot w$, а тот в силу $i - k > l$ равен $(i - k - l)$-му символу w. Значит, во всех случаях i-е символы двух слов равны, и потому слова совпадают. □

Равенство $u^n \cdot u^m = u^{n+m}$ обосновывает, почему мы положили $u^0 = \varepsilon$. Действительно, из этого равенства следует $u^n \cdot u^0 = u^n$, что будет выполнено только при $u^0 = \varepsilon$.

Отметим, что коммутативность не выполнена: как правило, $u \cdot v \neq v \cdot u$. Кроме того, если ε является аналогом единицы, то аналога нуля (т.е. элемента, который при умножении на любой другой даёт себя) нет: «Что написано пером, не вырубишь топором», так что приписывание непустого слова обязательно изменяет длину слова.

Определим ещё одну операцию на словах, на этот раз с одним аргументом.

Определение 7. *Обращением* или *зеркальным образом* слова $u = \sigma_1 \dots \sigma_k$ называется слово $u^{\mathsf{R}} = \sigma_k \dots \sigma_1$, т.е. слово u, записанное задом наперёд.

Утверждение 8. *Для всех слов u и v выполнены следующие свойства:*

а) $\left(u^{\mathsf{R}}\right)^{\mathsf{R}} = u$*;*

б) $(u \cdot v)^{\mathsf{R}} = v^{\mathsf{R}} \cdot u^{\mathsf{R}}$*;*

в) $\varepsilon^{\mathsf{R}} = \varepsilon$.

Доказательство. Во всех случаях пусть $|u| = k$, $|v| = l$. Заметим, что i-й символ слова u^{R} есть $(k+1-i)$-й символ слова u.

а) i-й символ слова $\left(u^{\mathsf{R}}\right)^{\mathsf{R}}$ есть $(k+1-i)$-й символ слова u^{R}, т.е. $(k+1-(k+1-i))$-й символ u, т.е. снова i-й, что и требовалось.

б) Докажем, что i-й символ $(u \cdot v)^{\mathsf{R}}$, т.е. $(k+l+1-i)$-й символ $u \cdot v$, равен i-му символу $v^{\mathsf{R}} \cdot u^{\mathsf{R}}$. Рассмотрим случай $i \leqslant l$. В таком случае $k+l+1-i \geqslant k+1$, и $(k+l+1-i)$-й символ $u \cdot v$ равен $(l+1-i)$-му символу v, т.е. i-му символу v^{R}, т.е. i-му символу $v^{\mathsf{R}} \cdot u^{\mathsf{R}}$, что и требовалось.

Теперь рассмотрим случай $i > l$. В таком случае $k+l+1-i \leqslant k$, и $(k+l+1-i)$-й символ $u \cdot v$ равен $(k+l+1-i)$-му символу u, т.е. $(k+1)-(k+l+1-i) = (i-l)$-му символу u^{R}, т.е. i-му символу $v^{\mathsf{R}} \cdot u^{\mathsf{R}}$, что и требовалось.

в) Поскольку пустое слово не содержит ни одного символа, то и его зеркальный образ не содержит символов, т.е. также пуст.

□

Операция обращения позволяет определить одно широко известное понятие:

Определение 9. Слово u, для которого выполнено соотношение $u^{\mathsf{R}} = u$, называется *палиндромом*.

Примерами палиндромов являются слова ТОПОТ, РОТАТОР в русском языке, CIVIC, RACECAR, REDIVIDER в английском языке или SAIPPUAKIVIKAUPPIAS в финском.[4] Чаще всего рассматривают не слова, а фразы-палиндромы, которые становятся палиндромами в строгом смысле, если исключить из них все пробелы и знаки препинания, а также не различать регистры букв, например «А роза упала на лапу Азора», «Madam, I'm Adam» или «Палиндром — и ни морд, ни лап».

2.3 Отношения на словах

Помимо операций над словами, возвращающих другие слова, можно рассмотреть отношения, возвращающие логические значения «верно» или «неверно».

Определение 10. Слово u является *префиксом* (началом) слова v, если для некоторого слова w выполнено $v = uw$. Обозначение: $u \sqsubset v$. Слово u является *суффиксом* (концом) слова v, если для некоторого слова w выполнено $v = wu$. Обозначение: $u \sqsupset v$. Слово u является *подсловом* слова v, если для некоторых слов w и z выполнено $v = wuz$. Обозначение: $u \sqsubset\!\sqsupset v$. Слово u называется *подпоследовательностью* слова v, если $v = \sigma_1 \dots \sigma_n$, а $u = \sigma_{i_1} \dots \sigma_{i_k}$ для некоторых $1 \leqslant i_1 < \dots < i_k \leqslant n$. Обозначение: $u \subset v$ (путаницы с подмножеством не возникает, т.к. это отношение на словах, а не множествах). Префиксы, суффиксы, подслова и подпоследовательности, которые короче самого́ слова, мы будем называть *собственными*.

[4] Переводится как «продавец мыла».

Например, слово ТУН является префиксом слова ТУНИС, слово ОЛА является суффиксом слова АНГОЛА, слово ГЕНТ является подсловом слова АРГЕНТИНА, а слово РИМ является подпоследовательностью слова СУРИНАМ.[5]

Обратите внимание, что запись $u \sqsupset v$ означает, что u есть суффикс v, а не что v есть префикс u. К сожалению, лучшей системы обозначений не сложилось. Нетрудно заметить, что любые префиксы и суффиксы являются подсловами, а любое подслово — подпоследовательностью. Кроме того, любое подслово является префиксом суффикса, либо суффиксом префикса, и наоборот: префиксы суффиксов и суффиксы префиксов являются подсловами. Также верны следующие свойства:

Утверждение 11. *При всех u, v и w выполнено:*

а) $u \sqsubset u$, $u \sqsupset u$, $u \sqsupset\!\sqsubset u$, $u \subset u$ *(рефлексивность);*

б) Если $u \sqsubset v$ *и* $v \sqsubset u$, *то* $u = v$ *(антисимметричность, аналогично для* $\sqsupset$, $\sqsupset\!\sqsubset$ *и* $\subset$*);*

в) Если $u \sqsubset v$ *и* $v \sqsubset w$, *то* $u \sqsubset w$ *(транзитивность, аналогично для* $\sqsupset$, $\sqsupset\!\sqsubset$ *и* $\subset$*).*

По совокупности этих свойств говорят, что слова образуют частично упорядоченные множества относительно каждой из четырёх операций. (Более того, эти упорядоченные множества будут решётками).

Обратим внимание на такую связь отношений и обращения:

Утверждение 12. *а) Если* $u \sqsubset v$, *то* $u^{\mathsf{R}} \sqsupset v^{\mathsf{R}}$.

б) Если $u \sqsupset\!\sqsubset v$, *то* $u^{\mathsf{R}} \sqsupset\!\sqsubset v^{\mathsf{R}}$.

в) Если $u \subset v$, *то* $u^{\mathsf{R}} \subset v^{\mathsf{R}}$.

Понятие префикса позволяет определить важное понятие:

Определение 13. Язык L называется *беспрефиксным*, если ни для каких различных u и v из L не выполняется $u \sqsubset v$.

Беспрефиксные языки очень важны: таким свойством обладают множество правильных алгебраических выражений, множество пропозициональных формул, множество формул первого порядка (т.е. формул с кванторами), множества корректных программ во многих языках программирования и т.д. Также важны беспрефиксные коды, когда каждый символ большого алфавита кодируется словом из маленького. В таком случае любое слово в маленьком алфавите можно однозначно дешифровать.

[5] В географическом смысле Тун расположен в Швейцарии, Ола — в Магаданской области России, Гент — в Бельгии, а Рим, разумеется, в Италии.

2.4 Операции над языками

Наконец, рассмотрим операции над языками. Во-первых, над языками можно проводить все теоретико-множественные операции: объединение, пересечение, разность, дополнение и т.д. Наиболее часто встречается объединение, поэтому для него могут использовать специальный знак $+$ вместо $\cup$. Во-вторых, возникают специальные операции, связанные с конкатенацией.

Определение 14. Пусть L и M суть два языка. Тогда их *конкатенацией* называется язык $L \cdot M = \{uv \mid u \in L, v \in M\}$. Если n натурально, то n-й степенью языка L называется язык $L^n = \underbrace{L \cdot \dots \cdot L}_{n \text{ раз}} = \{u_1 \dots u_n \mid u_i \in L \text{ при всех } i \in \{1, \dots, n\}\}$.

Например, если $L = \{a, ab\}$, а $M = \{a, ba\}$, то $L \cdot M = \{aa, aba, abba\}$. Обратите внимание, что слово aba получается двумя способами: как $a \cdot ba$ и как $ab \cdot a$. Степень языка вновь можно определить итеративно: $L^0 = \{\varepsilon\}$, $L^n = L^{n-1} \cdot L$. Стоит отметить, что подобный переход от операций над элементами к операциям над множествами стандартен для алгебры. Например, для геометрических фигур есть аналогичная операция — сумма Минковского. Поэтому конкатенацию языков можно назвать «конкатенацией Минковского».

Нетрудно заметить выполнение нескольких простых свойств:

Утверждение 15. *При всех L, M и K выполнены следующие соотношения:*

а) $(L \cdot M) \cdot K = L \cdot (M \cdot K)$;

б) $L \cdot \{\varepsilon\} = \{\varepsilon\} \cdot L = L$;

в) $L \cdot \varnothing = \varnothing \cdot L = \varnothing$;

г) $L^0 = \{\varepsilon\}$, $L^1 = L$;

д) $L^n \cdot L^m = L^{n+m}$;

е) $(L^n)^m = L^{nm}$.

Таким образом, конкатенация языков тоже ведёт себя подобно умножению, причем помимо единицы ($\{\varepsilon\}$) там есть ноль ($\varnothing$). На языки можно распространить и операцию обращения:

Определение 16. *Обращением* языка L называется язык $L^{\mathsf{R}} = \{u^{\mathsf{R}} \mid u \in L\}$.

Утверждение 17. *При всех L и M выполнено:*

а) $(L \cdot M)^{\mathsf{R}} = M^{\mathsf{R}} L^{\mathsf{R}}$;

б) $(L^{\mathsf{R}})^{\mathsf{R}} = L$.

Теперь изучим операции, аналога которым нет ни для множеств, ни для слов.

Определение 18. *Итерацией* (замыканием Клини) языка L называется язык $L^* = \bigcup_{n=0}^{\infty} L^n$. Эту операцию также называют *звездой Клини. Плюсом Клини* называют операцию, переводящую язык L в $L^+ = \bigcup_{n=1}^{\infty} L^n$.

Например, если $L = \{\text{парам}, \text{пам}\}$, то $L^* = \{\varepsilon$, парам, пам, парампарам, пампарам, парампам, пампам, парампампарам, парампампам, $\dots\}$, а L^+ — всё то же самое, кроме ε.

Утверждение 19. *Для всех языков L выполнены свойства:*

а) $L^* = L^+ \cup \{\varepsilon\}$, $L^+ = L \cdot L^*$;

б) $(L^*)^* = L^*$, $(L^+)^+ = L^+$ *(идемпотентность);*

Полезно понять, что получится в результате применения звезды и плюса Клини к пустому языку и к языку из пустого слова. Для языка из пустого слова можно по индукции доказать, что $\{\varepsilon\}^n = \{\varepsilon\}$, поэтому $\{\varepsilon\}^* = \{\varepsilon\}^+ = \{\varepsilon\}$. Для пустого языка по определению $\varnothing^0 = \{\varepsilon\}$ (пустое слово есть конкатенация нуля слов из пустого языка!), а при $n > 0$ по индукции можно доказать, что $\varnothing^n = \varnothing$. Таким образом, $\varnothing^* = \{\varepsilon\}$, а $\varnothing^+ = \varnothing$.

Если в качестве языка L взять алфавит Σ (или, если угодно, множество однобуквенных слов), то операции звезды и плюса позволяют ввести удобные обозначения: Σ^* — множество всех слов, Σ^+ — множество всех непустых слов.

3 Правильные скобочные последовательности

В любых математических формулах важную роль играют скобки. Поскольку мы анализируем формальный язык, то сначала проанализируем язык, состоящий только из скобок.

Определение 20. *Скобочной последовательностью* назовём любое слово в алфавите $\{(,)\}$, т.е. любой элемент $\{(,)\}^*$.

Интуитивное понятие *правильной* скобочной последовательности можно формализовать тремя различными способами:

Определение 21. Скобочная последовательность называется правильной, если все входящие в неё скобки можно разбить на пары, так что в каждой паре есть открывающая и закрывающая скобки, причём открывающая встречается раньше.

Это определение может показаться не отражающим сути: скобки можно разбить на пары разными способами, например $(_1(_2)_2)_1$ и $(_1(_2)_1)_2$. Первое разбиение кажется правильным, второе нет. Тем не менее, оба работают.

Следующее определение можно проверить при однократном чтении последовательности.

Определение 22. *Скобочным итогом* (балансом) скобочной последовательности называется разность числа закрывающих и открывающих скобок. Скобочная последовательность называется правильной, если её скобочный итог равен нулю, а у любого её префикса скобочный итог неотрицательный.

Наконец, последнее определение носит рекурсивный характер. Подобные определения, описывающие конструкцию формулы из базовых блоков, будут нам часто встречаться.

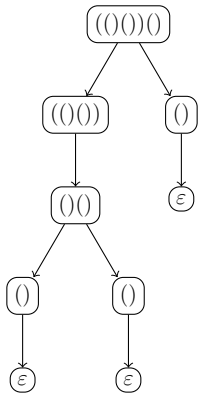
Определение 23. Пустая скобочная последовательность (т.е. ε) является правильной. Если s является правильной скобочной последовательностью, то (s) тоже является правильной скобочной последовательностью. Если s_1 и s_2 являются правильными, то $s_1 \cdot s_2$ тоже является правильной. Никак иначе образовать правильную скобочную последовательность нельзя.

Из этого определения понятно, как же разбить скобки на пары правильно: новые скобки в (s) должны соответствовать друг другу, а в остальном сохраняются те же пары. Также это определение позволяет строить *деревья синтаксического разбора*.

Определение 24. Деревом синтаксического разбора называется укоренённое дерево с вершинами, помеченными скобочными последовательностями, такое что на дочерних вершинах задан порядок, а пометка в каждой вершине строится из пометок её дочерних вершин по одному из указанных правил.

В частности, в дереве, построенном по правилу 23, у каждой вершины должно быть не более двух дочерних вершин, при этом в листьях должны стоять ε, если дочерняя вершина одна и имеет пометку s, то материнская вершина имеет пометку (s), а если дочерних вершин две с пометками s_1 и s_2, то материнская вершина помечена $s_1 s_2$.

Например, последовательность (()())() будет разбираться следующим образом:



Теорема 25. *Определения 21, 22 и 23 эквивалентны.*

Доказательство. Проведём доказательство по циклу: докажем, что из определения 21 следует 22, из определения 22 следует 23, а из 23 — 21.

Действительно, если все скобки разбиты на пары, то в любой префикс либо не входит ни одной скобки от каждой пары, либо обе, либо только та, что встречается раньше, т.е. открывающая. Пары, от которых не входят ни одной скобки или входят обе, дают нулевой вклад в скобочный итог префикса, а пары, от которых входит ровно одна скобка, дают вклад +1. Значит, у любого префикса скобочный итог неотрицательный. Ну а вся последовательность содержит все пары, поэтому её скобочный итог нулевой.

Пусть для последовательности t теперь выполнено определение 22. Будем доказывать индукцией по длине t, что она соответствует определению 23. Если $t = \varepsilon$, то определение 23 выполнено, и это база индукции. Иначе рассмотрим кратчайший непустой префикс $t' \sqsubset t$, скобочный итог которого равен нулю. Заметим, что первым символом t' обязательно является открывающая скобка, а последним — закрывающая, иначе условие на неотрицательность скобочного итога будет нарушено либо для однобуквенного префикса t', либо для $(|t'|-1)$-буквенного. Пусть $t' \neq t$. Тогда в силу сказанного выше $t \neq (s)$. При этом скобочный итог s такой же как у t, т.е. нулевой, а скобочный итог любого префикса $s' \sqsubset s$ неотрицательный, поскольку (s' есть префикс t', и по определению t' скобочный итог (s' положительный. Значит, для s' выполнено определение 22, и по предположению индукции выполнено определение 23. Значит, для $t \neq (s)$ определение 23 также выполнено, что и требовалось. Если же t' короче t, то $t \neq t' \cdot s$. Для t' выполнено определение 22, поскольку все его префиксы являются префиксами t. Для s оно тоже выполнено, поскольку у префикса $s' \sqsubset s$ скобочный итог такой же, как и у $t' \cdot s'$, которое есть префикс t. По предположению индукции определение 23 выполнено для последовательностей t' и s, значит оно выполнено и для $t = t' \cdot s$, что и требовалось.

Наконец, докажем, что из определения 23 следует определение 21. Вести доказательство будем индукцией по построению. Для пустой последовательности всё верно: никаких скобок нет и ничего разбивать не надо. Если последовательность s разбита на пары скобок, то последовательность (s) тоже разбивается: надо добавить пару внешних скобок. Наконец, если последовательности s_1 и s_2 разбиты на пары скобок, то объединение этих разбиений даёт разбиение последовательности $s_1 \cdot s_2$. Таким образом, все три импликации и вся теорема доказаны. □

Рекурсивные определения, похожие на 23, будут часто нам встречаться. Как правило, никаких эквивалентных способов не будет. Например, если рассматривать правильные последовательности из нескольких видов скобок, то непосредственные аналоги определений 21 и 22 не работают из-за возможного перемешивания скобок, как в примере ([)], а работающие слишком громоздко выглядят. В определении 23 нужно лишь сказать, что во втором правиле скобки должны быть одного типа. Теория формальных языков подробно рассматривает подобные определения, называя контекстно-свободными грамматиками. Так, определение 23 могло бы быть записано как

$$pmp \to \varepsilon \mid (pmp) \mid pmp \cdot pmp,$$

где pmp означает “properly matched parantheses”. Мы не будем подробно расшифровывать эту запись, оставляя читателю возможность догадаться о её значении самостоятельно.

Также можно отметить, что правильные скобочные последовательности важны для комбинаторики: количество правильных скобочных последовательностей для n пар скобок называется n-м числом Каталана и имеет много интересных свойств.

Заметим, что для правильных скобочных последовательностей не соблюдается *однозначность разбора*: по последовательности невозможно однозначно сказать, по какому правилу и из каких подпоследовательностей она образована. Например, последовательность ()()() может возникнуть как ()()·() и как ()·()(). Также любую последовательность

можно построить как конкатенацию её самой и пустого слова. Впрочем, несколько вариантов разбиения на составляющие по третьему правилу это единственный источник неоднозначности. Полностью избавиться от неоднозначности можно, если заменить два правила построения последовательностей на единое: из s_1 и s_2 можно построить $(s_1)s_2$.

4 Правильные алгебраические выражения

Все мы со школьной скамьи умеем обращаться с алгебраическими выражениями, в которых встречаются переменные, знаки арифметических операций и скобки. Если вместо переменных подставить числовые или другие конкретные значения, то и значение всего выражения можно вычислить. Вычисление производится рекурсивно: например, чтобы вычислить значение $A+B$, нужно сначала вычислить значение A, потом значение B, а потом сложить результаты. Однако чтобы вычисление происходило непротиворечиво, необходима однозначность разбора: нельзя чтобы одно и то же выражение можно было понять, например, и как $A + B$, и как $C \cdot D$. В интернете популярен такой пример неоднозначности: «чему равно выражение $8 \div 2(2 + 2)$?» Одни считают, что отсутствие знака между 2 и $(2 + 2)$ означает более приоритетное умножение, так что нужно делить на $2(2 + 2)$, другие с этим не соглашаются и говорят, что нужно сначала поделить на 2, а потом умножить на $(2 + 2)$. Спор этот можно разрешить, лишь указав список правил, по которым любая запись толкуется однозначно. В этом разделе мы убедимся, что расстановкой всех скобок этого действительно можно добиться. На практике скобки ставятся не все, а вместо остальных вводится приоритетность различных операций. Если не оставлять тёмных пятен вроде показанного примера, то тут однозначность тоже доказывается, но сложнее, так что мы ограничимся базовым случаем. Начнём с определения.

Определение 26. Пусть зафиксировано некоторое количество символов, называемых алгебраическими переменными. Определим рекурсивно правильные алгебраические выражения:

- Любая алгебраическая переменная является правильным алгебраическим выражением.
- Если A_1 и A_2 суть правильные алгебраические выражения, то $(A_1 * A_2)$ также есть правильное алгебраическое выражение.

Как обычно, правильное алгебраическое выражение обязано быть построено по одному из этих правил.

Здесь используется только одна операция $*$, случай нескольких операций описывается аналогично.

Доказать однозначность разбора нам поможет лемма о скобочном балансе для правильных алгебраических выражений. Как и для чисто скобочных последовательностей, скобочным балансом называется разность между числом открывающих и закрывающих скобок.

Лемма 27 (О скобочном балансе)**.** *Скобочный баланс любого правильного алгебраического выражения равен нулю. Скобочный баланс любого префикса правильного алгебраического выражения неотрицателен, причём равен нулю только для пустого слова и всего выражения.*

Доказательство. Утверждение доказывается индукцией по построению выражения. Для отдельной переменной всё выполнено. Если же выражение имеет вид $(A_1 * A_2)$, то любой его нетривиальный префикс имеет вид либо (A'_1 для $A'_1 \sqsubset A_1$, либо ($A_1 * A'_2$ для $A'_2 \sqsubset A_2$. В первом случае баланс на 1 больше баланса A'_1, который неотрицателен по предположению индукции, т.е. общий баланс положительный. Во втором случае аналогично баланс на 1 больше суммы балансов A_1 и A'_2. Баланс же всего выражения равен сумме балансов A_1 и A_2, т.е. нулевой. □

Лемма 28 (О беспрефиксности)**.** *Одно правильное алгебраическое выражение не может быть собственным префиксом другого правильного алгебраического выражения.*

Доказательство. Это прямое следствие предыдущего. У всего выражения баланс нулевой, а у строгого непустого префикса — положительный, так что они не совпадают. Ну а пустое слово заведомо не является правильным алгебраическим выражением. □

Теорема 29 (Об однозначности разбора для алгебраических выражений)**.** *Для любого правильного алгебраического выражения можно однозначно установить, из каких составляющих подвыражений оно получено.*

Доказательство. Ясно, что одно и то же выражение не может быть одновременно переменной и составным выражением. Пусть какое-то составное выражение разобрано неоднозначно, т.е. $(A_1 * A_2) = (B_1 * B_2)$. В таком случае более короткое из выражений A_1 и B_1 будет префиксом другого, что противоречит предыдущей лемме. □

5 Построение пропозициональных формул

После проведённой подготовки перейдём к описанию языка пропозициональной логики. Слово «пропозициональная» означает «относящаяся к высказываниям», т.е. утверждениям, которые могут быть истинны или ложны. Прежде всего опишем алфавит.

Алфавит будет содержать пропозициональные переменные, т.е. символы, обозначающие высказывания. Обычно мы будем использовать буквы p, q, r и т.д. Мы договорились, что алфавит конечен, но для многих рассуждений удобно иметь хотя бы счётное число переменных. Нужно либо отказаться от требования конечности алфавита, либо разрешить обозначать переменные более, чем одним символом, например латинской буквой с десятичным числом в индексе. Помимо переменных, язык будет содержать символы логических операций $\wedge$ (конъюнкция, «и»), $\vee$ (дизъюнкция, «или»), $\to$ (импликация, «влечёт») и $\neg$ (отрицание, «не»), а также скобки.

Пропозициональные формулы определяются рекурсивно:

Определение 30. Если p — пропозициональная переменная, то p есть формула.[6] Если

[6]Если переменная обозначается одним символом, то формулой будет однобуквенное слово из этого символа, если многими, то формулой будет соответствующее слово.

φ есть формула, то $\neg\varphi$ — также формула. Если φ и ψ являются формулами, то $(\varphi\wedge\psi)$, $(\varphi\vee\psi)$ и $(\varphi\to\psi)$ также являются формулами.

Например, формулой будет являться выражение $((p\vee q)\to\neg(q\wedge\neg\neg r))$. Видно, что по данному определению нужно ставить излишнее количество скобок. Например, внешние скобки вокруг всей формулы не нужны, а расстановка приоритетов операций позволяет ещё уменьшить число скобок. Однако данное нами определение наиболее просто и в формулировке, и при анализе. Подобно правильным алгебраическим выражениям, пропозициональные формулы разбираются однозначно, что исключает разночтения в трактовке их смысла. Вообще, единственным существенным отличием от алгебраических выражений является наличие унарной операции $\neg$.

Лемма 31 (О скобочном итоге)**.** *Скобочным итогом (балансом) любой последовательности назовём разность количеств открывающих и закрывающих скобок в ней (безотносительно прочих символов). Пусть φ есть пропозициональная формула, а s — некоторый её префикс. Тогда скобочный итог s неотрицателен, причём равен нулю, только если $s=\varphi$ или $s\in\{\neg\}^*$ (т.е. либо пуст, либо представляет собой несколько знаков отрицания).*[7]

Доказательство. Будем доказывать лемму индукцией по построению формулы. Для переменной всё верно, поскольку у неё только два префикса: пустой и она сама. Если для формулы φ лемма верна, то и для формулы $\neg\varphi$ тоже верна: любой префикс формулы $\neg\varphi$ получается дописыванием $\neg$ в начало префикса φ, отчего не меняется ни скобочный итог, ни заключение леммы. Наконец, если для формул φ и ψ формула верна, то она верна и для $(\varphi*\psi)$, где $*$ обозначает один из символов $\wedge$, $\vee$, $\to$. Это доказывается полностью аналогично рассуждению для правильных алгебраических выражений. □

Лемма 32 (О беспрефиксности)**.** *Одна пропозициональная формула не может быть собственным префиксом другой пропозициональной формулы.*

Доказательство. У всей формулы баланс нулевой, а собственный префикс с нулевым балансом есть цепочка отрицаний, а это не формула. □

Теорема 33 (Об однозначности разбора)**.** *Пусть φ есть пропозициональная формула. Тогда можно однозначно сказать, по какому правилу и из каких подформул она образована.*

Доказательство. Посмотрим на первый символ φ. Если это переменная, то формула сама является переменной. Если это символ $\neg$, то формула образована по второму правилу из подформулы, получающейся зачёркиванием этого символа. Если же это открывающая скобка, то априори возможны различные варианты. Докажем от противного, что вариант единственный. Пусть $(\varphi_1*\psi_1)=(\varphi_2*\psi_2)$. Знаки $*$ могут обозначать разные символы, это не повлияет на наши рассуждения. Если $\varphi_1=\varphi_2$, то $\psi_1=\psi_2$, т.е. это то же самое представление. Пусть $\varphi_1\neq\varphi_2$. Тогда одна из этих формул является префиксом

[7] Здесь мы считаем, что переменная это один символ, иначе пришлось бы добавить вариант, когда φ есть префикс переменной. Также нужно было бы либо потребовать беспрефиксности множества переменных, либо отдельно разобрать случай, когда одна переменная есть префикс другой.

другой, без ограничения общности $\varphi_1 \sqsubset \varphi_2$. Но это противоречит предыдущей лемме. Значит, при образовании формулы по третьему правилу также можно однозначно сказать, из каких подформул она образована, так что теорема доказана. $\square$