

12

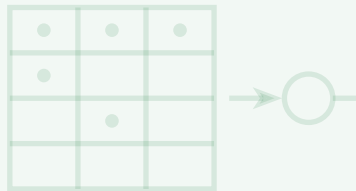
Лекция

24–25 ноября

Булевы функции и SAT

Константин Чухарев

Дискретная математика — Осень 2026



Булевы функции

Логика — алгебра высказываний.

— Джордж Буль

От логики к железу

Джордж Буль в 1854 году алгебраизировал логику: конъюнкция стала умножением, дизъюнкция — сложением, отрицание — дополнением.

В 1937 году Клод Шеннон показал, что та же алгебра описывает поведение реле: замкнутая цепь — истина, разомкнутая — ложь.

Логика XIX века стала железом XX века: каждая булева формула — цепь переключателей.

План лекции: научиться представлять функцию формулой (нормальные формы) и проверять формулу перебором (задача SAT).

Булева функция

Определение 1

Булева функция: $f : \{0, 1\}^n \rightarrow \{0, 1\}$.

Аргументы и значение — биты.

Функция полностью задаётся таблицей истинности из 2^n строк.

«Задать функцию» и «заполнить таблицу» — одно и то же: никакого другого описания произвольной булевой функции нет.

Таблица истинности

Пример

$$f(x, y) = x \wedge y:$$

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Таблица — полное, но громоздкое описание: строк 2^n .

Формула и схема — компактные описания для функций, у которых структура есть.

Сколько булевых функций?

Теорема 1 (Подсчёт)

Различных булевых функций от n переменных ровно 2^{2^n} .

Доказательство

Докажем подсчётом таблиц истинности.

В таблице 2^n строк, и выход в каждой строке выбирается независимо: 0 или 1.

Значит, таблиц 2^{2^n} , и каждая задаёт свою функцию.

Две шкалы роста

строк таблицы: 2^n , булевых функций: 2^{2^n} .

Пример

- $n = 2$: $2^4 = 16$ — все классические вентили.
- $n = 5$: $2^{32} > 4$ миллиарда.
- $n = 10$: 2^{1024} — больше, чем атомов в наблюдаемой вселенной.

Рост 2^{2^n} быстрее любой экспоненты: таблица — полное, но безнадежно громоздкое описание функции.

Бинарных функций шестнадцать

При $n = 2$ у каждой из шестнадцати функций есть имя.

Имя	Формула	Смысл
Конъюнкция	$x \wedge y$	AND
Дизъюнкция	$x \vee y$	OR
Отрицание импликации	$\neg(x \rightarrow y)$	антиимпликация
Обратная импликация	$y \rightarrow x$	обратная
XOR	$x \oplus y$	входы различны
Эквивалентность	$x \equiv y$	входы совпадают
Штрих Шеффера	$x \uparrow y$	NAND
Стрелка Пирса	$x \downarrow y$	NOR

Остальные шесть — проекции x и y , их отрицания и константы 0, 1.

Нормальные формы: напоминание

В модуле о логике каждая формула приводилась к каноническому виду.

- ДНФ — дизъюнкция конъюнкций литералов, КНФ — конъюнкция дизъюнкций.

Каждая дизъюнкция литералов — **дизъюнкт**.

- Совершенные формы СДНФ и СКНФ единственны — канонический портрет функции.

Здесь тот же объект глазами инженера: формула как проект схемы.

Минтерм

Определение 2 (Минтерм)

Минтерм — конъюнкция литералов (переменных и их отрицаний), в которую каждая переменная входит ровно один раз.

Конъюнкция истинна лишь когда истинны все её литералы.

Поэтому минтерм равен 1 ровно на одной строке таблицы — и 0 на всех остальных.

Макстерм

Определение 3 (Макстерм)

Макстерм — дизъюнкция литералов, в которую каждая переменная входит ровно один раз.

Дизъюнкция ложна лишь когда ложны все её литералы.

Поэтому макстерм равен 0 ровно на одной строке — и 1 на всех остальных.

Детекторы строк

Пример

При $n = 3$ минтерм $x \wedge \bar{y} \wedge z$ истинен только на наборе $(1, 0, 1)$.

Макстерм $x \vee \bar{y} \vee z$ ложен только на наборе $(0, 1, 0)$.

Минтерм — детектор единичной строки, макстерм — детектор нулевой.

Каждый однозначно соответствует строке таблицы истинности.

Нормальные формы XOR

Пример

$x \oplus y$ истинна на наборах $(0, 1)$ и $(1, 0)$, ложна на $(0, 0)$ и $(1, 1)$.

СДНФ — по единицам:

$$(\bar{x} \wedge y) \vee (x \wedge \bar{y}).$$

СКНФ — по нулям:

$$(x \vee y) \wedge (\bar{x} \vee \bar{y}).$$

Совершенная форма избыточна

Пример

Импликация $x \rightarrow y$ истинна на трёх наборах из четырёх:

$$x \rightarrow y = (\bar{x} \wedge \bar{y}) \vee (\bar{x} \wedge y) \vee (x \wedge y).$$

Алгебраическое упрощение сжимает совершенную форму до двух литералов вместо шести:

$$x \rightarrow y = \bar{x} \vee y.$$

СДНФ единственна, но избыточна: каждый лишний литерал — лишний вентиль.

Найти самую короткую равносильную ДНФ — задача минимизации, тема следующей лекции.



Задача SAT

Нет царского пути в геометрию.

— Евклид

Задача SAT

Определение 4 (Выполнимость)

Формула в КНФ **выполнима** (*satisfiable*), если существует набор значений переменных, при котором она истинна.

Задача SAT: по данной КНФ узнать, выполнима ли она.

Пример

$$\varphi = (x \vee \bar{y}) \wedge (\bar{x} \vee y \vee z) \wedge \bar{z}.$$

Набор $x = 1, y = 1, z = 0$ выполняет все три дизъюнкта.

Выполнимость и общезначимость

Выполнимость — не общезначимость: x выполнима при $x = 1$, но тавтологией не является.

Пример

$x \wedge \bar{x}$ — простейшая невыполнимая формула: противоречие при любом наборе.

Для SAT естественна КНФ: решатели работают именно с конъюнкцией дизъюнктов.

Любую формулу можно привести к КНФ, а схема — это уже почти КНФ.

Перебор экспоненциален

Простейший алгоритм — перебрать все наборы:

n переменных $\rightarrow 2^n$ наборов.

Пример

- $n = 20$: около миллиона наборов — мгновенно.
- $n = 40$: около триллиона — часы счёта.
- $n = 300$: наборов больше, чем атомов во вселенной.

Перебор можно ужать отсечениями, но экспонента в худшем случае остаётся.

Вопрос «можно ли быстрее?» — центральный вопрос всей теории сложности.

Дизъюнкт как импликация

Дизъюнкт из двух литералов равносителен паре импликаций:

$$l_1 \vee l_2 \equiv \overline{l_1} \rightarrow l_2, \quad \overline{l_2} \rightarrow l_1.$$

Вторая импликация — контрапозиция первой.

Потерять её — потерять половину рёбер будущего графа.

Тройку литералов так разложить нельзя — граница между лёгким и трудным случаями SAT.

Граф импликаций

Определение 5 (Граф импликаций)

Вершины — все литералы: x и $\bar{x}$ для каждой переменной.

Рёбра — обе импликации каждого дизъюнкта.

Выполнимость превращается в вопрос о достижимости в графе.

Если истинность литерала влечёт истинность другого, в графе есть путь.

Критерий 2-SAT

Теорема 2 (Критерий)

Формула 2-SAT невыполнима тогда и только тогда, когда x и $\bar{x}$ лежат в одной компоненте сильной связности графа импликаций.

Доказательство

Докажем в обе стороны.

Если x и $\bar{x}$ взаимно достижимы, импликации дают $x \rightarrow \bar{x}$ и $\bar{x} \rightarrow x$ — противоречие при любом значении x .

Если же пары разнесены по компонентам, присваивание по обратному топологическому порядку компонент делает каждую импликацию истинной.

К компонентам сильной связности приводит обход в глубину за $O(n + m)$.

Построение выполняющего набора

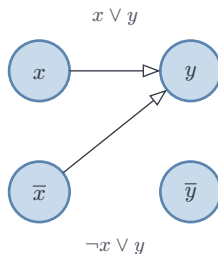
Если пары разнесены по компонентам, модель читается из графа.

1. Найти компоненты сильной связности и топологический порядок их конденсации.
2. Обработать компоненты в обратном порядке — от стоков к истокам.
3. Переменная получает 1, если её компонента идёт позже компоненты отрицания, иначе 0.

По пути в графе значение не убывает: ложь перед истиной.

Поэтому каждая импликация оказывается истинной.

Граф импликаций: пример



Пример

Формула $(x \vee y) \wedge (\bar{x} \vee y)$: оба дизъюнкта ведут в y .

Пары $x, \bar{x}$ и $y, \bar{y}$ лежат в разных компонентах — формула выполнима.

Из графа видно значение: оба пути требуют истинности y .

2-SAT против 3-SAT

2-SAT решается за линейное время.

3-SAT — SAT, где каждый дизъюнкт содержит ровно три литерала, — NP-полна.

Один лишний литерал — скачок сложности.

Причина структурная: пара литералов распадается на импликации, тройка — нет.

Длинный дизъюнкт режут на тройки свежими переменными (преобразование Цейтина), но в 2-SAT формула от этого не становится. Промышленные решатели берут общий SAT методом CDCL: на каждом конфликте учится новый дизъюнкт, а эвристика ведёт выбор переменных — так работают MiniSat и z3.

Кодирование в SAT: стратегия

1. Выбрать булевы переменные, описывающие решение.
2. Записать ограничения задачи дизъюнктами.
3. Запустить SAT-решатель.
4. Модель декодировать в решение, UNSAT — решения нет.

Хорошая кодировка наследует структуру задачи.

Плохая раздувает формулу и останавливает решатель.

Кодирование: раскраска графа

Задача: можно ли раскрасить вершины графа в k цветов так, чтобы смежные вершины различались?

Переменные: $x_{v,c} = 1$, если вершина v получает цвет c .

Ограничения:

- у каждой вершины есть цвет: $\bigvee_{c=1}^k x_{v,c}$.
- не более одного цвета: $\overline{x_{v,c}} \vee \overline{x_{v,d}}$ для каждой пары $c < d$.
- смежные различаются: $\overline{x_{u,c}} \vee \overline{x_{v,c}}$ для каждого ребра и цвета.

Всего $O(|V| \cdot k^2 + |E| \cdot k)$ дизъюнктов.

Треугольник в два цвета

Пример (Полный пример)

Граф — треугольник: вершины a, b, c попарно смежны, цветов $k = 2$.

Дизъюнкты «у каждой вершины есть цвет»:

$$(x_{a,1} \vee x_{a,2}) \wedge (x_{b,1} \vee x_{b,2}) \wedge (x_{c,1} \vee x_{c,2}).$$

Дизъюнкты рёбер — по одному на каждый цвет:

$$\begin{aligned} &(\overline{x_{a,1}} \vee \overline{x_{b,1}}) \wedge (\overline{x_{a,1}} \vee \overline{x_{c,1}}) \wedge (\overline{x_{b,1}} \vee \overline{x_{c,1}}) \wedge \\ &(\overline{x_{a,2}} \vee \overline{x_{b,2}}) \wedge (\overline{x_{a,2}} \vee \overline{x_{c,2}}) \wedge (\overline{x_{b,2}} \vee \overline{x_{c,2}}). \end{aligned}$$

Попарные запреты «не более одного цвета» опустим: для противоречия они не нужны.

Треугольник: вывод

Пример

Цветов два, вершин три: какие-то две вершины получили один цвет.

Эти две смежны, и их рёберный дизъюнкт запрещает совпадение.

Формула невыполнима — треугольник нельзя раскрасить в два цвета.

Ответ UNSAT — доказательство несуществования.

Ответ SAT — готовая раскраска: модель формулы читается как решение задачи.

Судоку и N-ферзи

- Судоку: $x_{r,c,d}$ — «в клетке (r, c) стоит цифра d ». $9^3 = 729$ переменных, около 12 тысяч дизъюнктов — решатель справляется за миллисекунды.
- N-ферзи: $x_{i,j}$ — «ферзь в клетке (i, j) ». При $N = 8$ — порядка семисот дизъюнктов.

Решатель не знает правил игры: ему дан плоский список дизъюнктов.

Хорошая кодировка делает структуру задачи частью самих дизъюнктов.

Лекция 12: что мы поняли?

- Булева функция — таблица из 2^n строк. Всего функций 2^{2^n} — рост быстрее любой экспоненты.
- Минтерм ловит одну единичную строку, макстерм — одну нулевую. Дизъюнкция минтермов даёт СДНФ, конъюнкция макстермов — СКНФ.
- Совершенная форма единственна, но избыточна — отсюда задача минимизации.
- SAT — вопрос о выполнимости КНФ. 2-SAT линеен на графе импликаций, 3-SAT NP-полна.
- Кодирование в SAT — переменные плюс ограничения-дизъюнкты: UNSAT доказывает несуществование, SAT даёт готовое решение.

Вопросы для самопроверки

1. Постройте СДНФ и СКНФ для $f(x, y) = x \rightarrow y$. Совпадает ли минимальная ДНФ с одной из совершенных форм?
2. Сколько булевых функций от трёх переменных? А от четырёх?
3. Переведите дизъюнкт $(\bar{x} \vee y)$ в пару импликаций и выпишите все рёбра графа импликаций для $(x \vee y) \wedge (\bar{x} \vee y)$.
4. Проверьте по графу импликаций выполнимость $(x \vee y) \wedge (\bar{x} \vee y) \wedge \bar{y}$.
5. Закодируйте в КНФ требование «ровно одна из переменных x, y, z истинна».
6. Почему дизъюнкт из трёх литералов не разлагается в пару импликаций?
7. Сколько наборов переберёт полный перебор при $n = 30$?

Чтение к следующей паре

- m10 — «Булевы функции», «ДНФ и КНФ».
- m14 — «Задача SAT», «Кодирование задач в SAT».
- Флеш-опросник — в начале лекции 13.