Определения и примеры

- Функция $f : \{0,1\}^n \to \{0,1\}$ — *n*-местная (*n*-арная) булева функция
  - будем писать $f(x_1, \ldots, x_n)$ или $f(\vec{x})$, если n известно или несущественно
  - также принято сокращать слова «булева функция» до б.ф.
- n-местная б.ф. f переводит строки из n бит в битовые значения, то есть
  - ⋆ задает n-местную операцию на множестве $\{0,1\}$
  - ⋆ вычисляет n-местный предикат на множестве $\{0,1\}$
  - ⋆ задает n-местное отношение на множестве $\{0,1\}$
  - ⋆ распознает язык $L_f \subseteq \{0,1\}^n$
- Прямолинейный (неэкономичный) способ задания б.ф. — таблица значений
  - также называемая таблицей истинности
  - ⋆ n-местную б.ф. f можно задать битовой строкой $F[0..2^n{-}1]$, где $F[i]$ — значение f на строке, являющейся двоичной записью числа i
- Пример: рассмотрим бинарные б.ф. $f(x,y)$, они же строки $F[0..3]$

x,y	0	$\wedge$	$>$	x	$<$	y	$+$	$\vee$	$\downarrow$	$\sim$	$\bar{y}$	$\leftarrow$	$\bar{x}$	$\to$	$'$	1
00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

  - $x \downarrow y = \overline{x \vee y}$ — стрелка Пирса, $x\,'\,y = \overline{x \wedge y}$ — штрих Шефера
  - $x > y, x < y$ — обычные бинарные отношения,
  - $x + y$ — сложение по модулю 2 (xor), $x \sim y = [x = y]$ — эквиваленция
  - $x \to y, x \leftarrow y$ — прямая и обратная импликации

Примеры многоместных булевых функций

Есть несколько важных серий n-местных булевых функций для произвольного n

- $\mathrm{PROJ}_i(\vec{x}) = x_i$ — проекции
  - ⋆ проекция вытаскивает нужный бит из вектора аргументов
  - чтобы вычислить $(a+b) \bmod 2$, где числа a и b заданы двоичными представлениями, нужно взять по младшему биту из a и b и выполнить xor
- $\bigoplus \vec{x}, \bigvee \vec{x}, \bigwedge \vec{x}$ — n-арные аналоги коммутативных бинарных функций
- $\mathrm{MOD}_p(\vec{x}) = [x_1 + \cdots + x_n$ делится на $p]$ — модулярные функции
  - ⋆ важный частный случай — MOD_2 (или PARITY) — четность числа единиц
  - ! постройте ДКА, который вычисляет функцию MOD_p
- $T_i(\vec{x}) = [x_1 + \cdots + x_n \geqslant i]$ — пороговые функции
  - ⋆ $T_1(\vec{x}) = \bigvee \vec{x}$, $T_n(\vec{x}) = \bigwedge \vec{x}$
  - ⋆ $T_{\lfloor n/2 \rfloor + 1}(\vec{x})$ — функции большинства (или голосования)
  - ⋆ пороговая функция — ключевой элемент персептрона (простейшей нейронной сети)

⋆ Функция $f(x_1, \ldots, x_n)$ может зависеть от значений только части переменных
  - ⋆ например, в списке бинарных функций есть x, $\bar{x}$, y, $\bar{y}$, 0, 1
  - переменная x_i функции f — фиктивная, если для любых $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ $f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) = f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$
  - переменная, не являющаяся фиктивной, называется существенной

Суперпозиция. Булевы формулы

- ⋆ Функции n переменных громоздко задавать таблицами; что делать?
- Сложные функции можно представлять как суперпозицию более простых
  - ⋆ рассматриваются функции нескольких переменных $\Rightarrow$ одни и те же функции могут образовывать много различных суперпозиций
  - $g(y_1, \ldots, y_k)$ можно подставить в $f(x_1, \ldots, x_n)$ вместо любого аргумента
  - какие-то из аргументов g можно отождествить с какими-то из аргументов f
  - пример: подстановкой $g(z,t) = z \wedge t$ в $f(x,y) = x \to y$ можно получить $(z \wedge t) \to y$, $(z \wedge y) \to y$, $x \to (z \wedge t)$, $x \to (x \wedge t)$
- Удобный способ записи булевых функций:
  - зафиксировать маленький набор функций B, смотреть на функции из B как на операции
  - записать булеву формулу — формальное выражение, содержащее переменные, символы операций из B и скобки
  - вычислять значение функции для каждого вектора значений переменных, выполняя операции
  - ⋆ булева формула $=$ слово над конечным алфавитом, которое удовлетворяет набору ограничений (синтаксис)
  - ⋆ булева формула задает булеву функцию (семантика)
- ⋆ Разные формулы могут задавать одну и ту же функцию
  - пример: $(x \vee y) \wedge z$ и $(x \wedge z) \vee (y \wedge z)$
  - формулы, задающие одну и ту же функцию, называются эквивалентными
  - тавтология — это формула, задающая константу 1
  - противоречие — это формула, задающая константу 0
  - формула, задающая функцию, отличную от константы 0, называется выполнимой

- Разные формулы задают одну функцию $\Rightarrow$ нужны «канонические» формулы
- Такие формулы называют нормальными формами
- Нормальная форма должна
  - существовать для любой функции
  - эффективно вычисляться (например по таблице истинности)
  - быть удобной для хранения и вычислений
- Литерал — это формула вида x или $\bar{x}$, где x — переменная
- Дизъюнктивная нормальная форма (ДНФ) — это формула вида $\bigvee_{i=1}^{n} F_i$, $n \geqslant 1$
  - где $F_i = \bigwedge_{j=1}^{k_i} L_{ij}$, L_{ij} — литерал, $k_i \geqslant 1$
  - F_i называют элементарными конъюнкциями
  - ⋆ ДНФ — иерархическая формула: отрицание применяется только к переменным, конъюнкция — к литералам, дизъюнкция — к элементарным конъюнкциям
- ДНФ от k переменных называется совершенной (k-СДНФ), если
  - все элементарные конъюнкции F_i различны
  - каждая F_i состоит из k литералов, соответствующих различным переменным

Теорема

Любая булева функция, не равная константе 0, задается некоторой СДНФ.

★ Иными словами, любая выполнимая формула эквивалентна СДНФ

★ Следствие: любая булева функция задается некоторой ДНФ

- $x \wedge \bar{x}$ — ДНФ, задающая 0

- Функцию $x \sim y$ иногда удобно записывать как x^y
  - ⋆ свойства: $x^1 = x$, $x^0 = \bar{x}$, $1^y = y$, $0^y = \bar{y}$
- Доказательство теоремы об СДНФ:
  - пусть $f(x_1, \ldots, x_k)$ отлична от константы 0
  - построим СДНФ F по следующему правилу:
  - ⋆ для каждого битового вектора $\vec{b} = (b_1, \ldots, b_k)$ такого, что $f(b_1, \ldots, b_k) = 1$, поместим в F элементарную конъюнкцию $C_{\vec{b}} = x_1^{b_1} \wedge \cdots \wedge x_k^{b_k}$
  - построенная формула — СДНФ по определению
  - докажем, что F задает f
  - ⋆ функция, заданная ДНФ, равна 1 $\Leftrightarrow$ одна из элементарных конъюнкций равна 1
  - ⋆ $f(b_1, \ldots, b_k) = 1 \Leftrightarrow C_{\vec{b}}(b_1, \ldots, b_k) = 1$ □

Пример построения СДНФ:

x_1, x_2, x_3	f
000	1
001	0
010	0
011	1
100	0
101	1
110	0
111	0

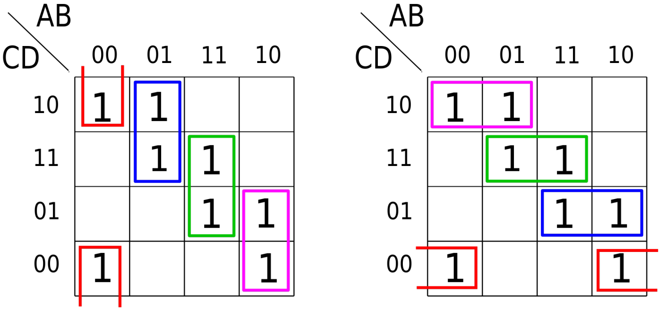
$$\Longrightarrow F = (\bar{x}_1 \wedge \bar{x}_2 \wedge \bar{x}_3) \vee (\bar{x}_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \bar{x}_2 \wedge x_3)$$

- ⋆ СДНФ — очень громоздкая формула
  - если у функции половина значений — единицы, то k-СДНФ состоит из $k \cdot 2^{k-1}$ литералов
  - ⋆ проще хранить 2^k бит таблицы значений
- ⋆ важная задача — построение кратчайшей ДНФ для данной функции
  - к сожалению, эта задача не только важная, но и трудная
- ⋆ Для б.ф. 2-4 переменных кратчайшую ДНФ строят при помощи карт Карно
- Карта Карно функции f — это специальная запись таблицы значений f
  - карта — это прямоугольная таблица из 2^k клеток, где k — арность f
  - строки и столбцы проиндексированы так, что каждой клетке однозначно соответствует набор значений переменных, в клетке пишется значение функции (обычно пишут только единицы)
  - ⋆ кратчайшую ДНФ строят, покрывая все клетки с единицами прямоугольниками такими, что
    - число клеток в прямоугольнике — степень двойки
    - если клеток 2^i, то $k-i$ переменных принимают в этих клетках одно значение (определяют элементарную конъюнкцию), а остальные i — все наборы значений

  пример:

$$(x \wedge \bar{z}) \vee (y \wedge \bar{z}) \vee (x \wedge \bar{y}) \Longrightarrow \begin{array}{c|c|c|c} & y & \bar{y} & \\ \hline x & & 1 & z \\ \hline x & 1 & 1 & \bar{z} \\ \hline \bar{x} & 1 & & \bar{z} \\ \hline \bar{x} & & & z \end{array} \Longrightarrow (y \wedge \bar{z}) \vee (x \wedge \bar{y})$$

- Еще одна проблема с кратчайшей ДНФ — она не единственна
- Вот иллюстрация из Википедии с картами Карно для 4 переменных:



$$(\bar{A} \wedge \bar{B} \wedge \bar{D}) \vee (\bar{A} \wedge B \wedge C) \vee (A \wedge B \wedge D) \vee (A \wedge \bar{B} \wedge \bar{C}) =$$
$$(\bar{A} \wedge C \wedge D) \vee (B \wedge C \wedge D) \vee (A \wedge \bar{C} \wedge D) \vee (\bar{B} \wedge \bar{C} \wedge \bar{D})$$

- Конъюнктивная нормальная форма (КНФ) — это формула вида $\bigwedge_{i=1}^{n} F_i$, $n \geqslant 1$
  - где $F_i = \bigvee_{j=1}^{k_i} L_{ij}$, L_{ij} — литерал, $k_i \geqslant 1$
  - F_i называют элементарными дизъюнкциями или клозами (clause)
  - ⋆ КНФ — тоже иерархическая формула: отрицание применяется только к переменным, дизъюнкция — к литералам, конъюнкция — к клозам
- КНФ от k переменных называется совершенной (k-СКНФ), если
  - все клозы F_i различны
  - каждый F_i состоит из k литералов, соответствующих различным переменным

Теорема

Любая булева функция, не равная константе 1, задается некоторой СКНФ.

★ Иными словами, любая не-тавтология эквивалентна СКНФ

★ Следствие: любая булева функция задается некоторой КНФ
  - $x \vee \bar{x}$ — КНФ, задающая 1

... симметрично доказательству для СДНФ

- Доказательство:
  - пусть $f(x_1,\ldots,x_k)$ отлична от константы 1
  - построим СКНФ F по следующему правилу:
  - ⋆ для каждого битового вектора $\vec{b}=(b_1,\ldots,b_k)$ такого, что $f(b_1,\ldots,b_k)=0$, поместим в F элементарную дизъюнкцию $D_{\vec{b}}=x_1^{\bar{b}_1}\vee\cdots\vee x_k^{\bar{b}_k}$
  - построенная формула — СКНФ по определению
  - докажем, что F задает f
  - ⋆ функция, заданная КНФ, равна 0 $\Leftrightarrow$ одна из элементарных дизъюнкций равна 0
  - ⋆ $f(b_1,\ldots,b_k)=0 \Leftrightarrow D_{\vec{b}}(b_1,\ldots,b_k)=0$ □

Пример построения СКНФ (функция — та же, что и для примера с СДНФ):

x_1,x_2,x_3	f
000	1
001	0
010	0
011	1
100	0
101	1
110	0
111	0

$$\Longrightarrow \quad F=(x_1\vee x_2\vee \bar{x}_3)\wedge(x_1\vee \bar{x}_2\vee x_3)\wedge(\bar{x}_1\vee x_2\vee x_3)\wedge \wedge(\bar{x}_1\vee \bar{x}_2\vee x_3)\wedge(\bar{x}_1\vee \bar{x}_2\vee \bar{x}_3)$$

★ Симметрия распространяется и на оптимизацию КНФ при помощи карт Карно: прямоугольники соответствуют клозам и покрывают множество нулей

Полные системы

- Множество B булевых функций называется полной системой, если формулой с множеством операций B можно задать любую булеву функцию
  - ⋆ от любого числа переменных, не меньшего 1
    пример: формула $x \wedge \bar{x}$ задает унарную функцию $f(x)$, равную 0
    пример: формула $x \vee y$ задает не только функцию $f(x, y)$ с таблицей значений

x, y	f
00	0
01	1
10	1
11	1

, но и функцию $g(x, y, z)$ с таблицей значений

x, y, z	g
000	0
001	0
010	1
011	1
100	1
101	1
110	1
111	1

★ Если все функции полной системы B можно задать формулами над множеством функций B', то B' — полная система

⋆ множество $\{\wedge, \vee, ^{-}\}$ является полной системой
  - например, по следствию из теоремы об СКНФ (предыдущий фрагмент)

! Какие еще полные системы существуют?
  - любое надмножество множества $\{\wedge, \vee, ^{-}\}$
    - включая множества всех булевых функций и всех бинарных булевых функций
  - множества $\{\wedge, ^{-}\}$ и $\{\vee, ^{-}\}$
    - выразить $\vee$ $(\wedge)$ через две оставшиеся функции по формулам де Моргана
    - сослаться на замечание ★

Напомним, что $x \downarrow y = \overline{x \vee y}$, $x\,'\,y = \overline{x \wedge y}$

Теорема

Множества $\{\downarrow\}$ и $\{\,'\,\}$ являются полными системами.

Доказательство:

- выразим отрицание и дизъюнкцию через стрелку Пирса:
  - ⋆ $\bar{x} = x \downarrow x$
  - ⋆ $x \vee y = \overline{x \downarrow y} = (x \downarrow y) \downarrow (x \downarrow y)$
- поскольку $\{\vee, \bar{\ }\}$ — полная система, $\{\downarrow\}$ — тоже полная
- аналогично, отрицание и конъюнкция выражаются через штрих Шефера □

- Верна и обратная теорема:
  - ★ если f — бинарная б.ф. и $\{f\}$ — полная система, то $f \in \{\downarrow, \,'\,\}$
  - обратная теорема следует из теоремы Поста о полноте (докажем потом)
- Еще одну полную систему рассмотрим в следующем фрагменте

Многочлены над $\mathbb{F}_2$

- Поле $\mathbb{F}_2$ — это множество $\{0,1\}$ с операциями $+$ (по $\bmod 2$) и $\cdot$ $(= \wedge)$
  - таблицы операций в привычном виде:

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

- ⋆ Многочлены от k переменных над $\mathbb{F}_2$ — это k-местные булевы функции
- ⋆ Множество функций $\{+,\cdot,1\}$ — полная система
  - из нее получается $\{\wedge,\bar{}\}$, так как $x\wedge y = xy$, $\bar{x} = x+1$

$\Rightarrow$ любую функцию можно записать формулой над $\{+,\cdot,1\}$

- Заметим, что
  - пользуясь коммутативностью и дистрибутивностью, можно раскрывать скобки и приводить подобные слагаемые
  - в $\mathbb{F}_2$ выполняются тождества $xx = x$ и $x+x=0$

$\Rightarrow$ Любая формула над $\{+,\cdot,1\}$ эквивалентна многочлену, в котором
  - каждый одночлен — это произведение переменных, в котором все переменные различны, либо свободный член (1 или 0)
  - все одночлены различны

- Описанный канонический вид многочлена называется полиномом Жегалкина
- ⋆ Для однозначности записи договоримся, что
  - алфавит переменных Σ — упорядочен
  - в каждом одночлене переменные записываются по возрастанию
  - одночлены записываются по возрастанию в радиксном порядке на Σ^*

Теорема

Любая булева функция задается полиномом Жегалкина, и притом единственным.

Доказательство:

- существование полинома следует из полноты системы $\{+, \cdot, 1\}$ и эквивалентности любой формулы над этой системой полиному Жегалкина (предыдущий слайд)

Единственность: зафиксируем алфавит переменных $\Sigma = \{x_1, \ldots, x_k\}$

- над этим алфавитом существует 2^{2^k} различных булевых функций
  - ⋆ таблица значений б.ф. задается битовым вектором длины 2^k
- одночлены над Σ биективно отображаются на подмножества Σ

$\Rightarrow$ существует 2^k различных одночленов над Σ

- полиномы Жегалкина над Σ биективно отображаются на множества одночленов
  - полиному 0 сопоставим пустое множество одночленов

$\Rightarrow$ существует 2^{2^k} различных полиномов Жегалкина над Σ

★ функция, которая каждому полиному Жегалкина над Σ ставит в соответствие задаваемую им б.ф. — сюръекция

- так как каждая функция задается полиномом Жегалкина

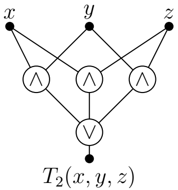
★ сюръекция между двумя конечными множествами одной мощности является инъекцией

- каждая функция задается единственным полиномом Жегалкина □

⋆ Полином Жегалкина — это нормальная форма («алгебраическая»)

- Булева схема (circuit) — альтернативный способ задания булевых функций
  - абстрагирует конструкцию электрической схемы из элементов (вентили, gates)
- Булеву функцию $f(x_1, \ldots, x_k)$ вычисляет черный ящик
  - у ящика k входящих проводов $(x_1, \ldots, x_k)$ и один выходящий (f)
  - ток, идущий по проводу, означает 1, отсутствие тока — 0
  - если токи во входящих проводах соответствуют вектору $(b_1, \ldots, b_k)$, то ток в выходном проводе кодирует $f(b_1, \ldots, b_k)$
- Внутри черного ящика находятся элементы, соединенные проводами в определенном порядке между собой, со входами и выходами
  - каждый элемент — это черный ящик, реализующий одну из функций полной системы B (базы)
  - в реальных электрических и электронных схемах элементы — это физические устройства, такие как реле в дверном звонке или диоды в электронных часах
  - мы рассматриваем идеальные элементы, абстрагируясь от физических сущностей

Пример: функцию большинства от трех переменных можно задать формулой $T_2(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$, которая представляется схемой



Булевы вектор-функции. Сложение столбиком

- Булева вектор-функция — это произвольная функция $\vec{f}: \{0,1\}^n \to \{0,1\}^m$
  - сложение двух n-битных чисел — это функция $ADD_n : \{0,1\}^{2n} \to \{0,1\}^{n+1}$
  - ⋆ вектор-функции намного удобнее задавать схемами, чем формулами
    - у схемы для вектор-функции m выходов вместо одного
- ⋆ Научимся вычислять функцию ADD_n
  - пусть $a = a_{n-1} \cdots a_0$, $b = b_{n-1} \cdots b_0$ — числа в двоичной записи
    - ведущие нули разрешены
  - $ADD_n(a_{n-1}, \ldots, a_0, b_{n-1}, \ldots, b_0) = (s_n, \ldots, s_0)$, где $s = s_n \cdots s_0 = a + b$
- ⋆ Вычисление столбиком:
  - пусть $c_n, \ldots, c_0$ — вспомогательные булевы переменные, c_i = перенос в разряд i
  - ⋆ $c_0 = 0$, $s_0 = a_0 + b_0$ (сложение по $\mod 2$!)
  - ⋆ $c_i = T_2(a_{i-1}, b_{i-1}, c_{i-1})$ для $i = 1, \ldots, n$ (почему?)
  - ⋆ $s_i = a_i + b_i + c_i$ для $i = 1, \ldots, n-1$; $s_n = c_n$
- Приведенный алгоритм выполняет $\Theta(n)$ операций
- Как и любой другой алгоритм сложения n-битных чисел

  ... но есть нюанс ...
- булева схема — это ациклический орграф
- электрический ток способен течь по проводам параллельно, давая возможность параллельного вычисления значений в разных узлах схемы (вершинах графа)
  - ★ время вычисления функции булевой схемой определяется глубиной схемы — максимальной длиной пути от входа до выхода
- Глубина схемы, построенной по алгоритму сложения столбиком, равна $\Theta(n)$
  - ⋆ никакой выгоды от распараллеливания мы не получаем

Параллельная схема для сложения

- Проблема сложения столбиком — в последовательном вычислении переносов
  - если все переносы известны, все биты s_i вычисляются параллельно за один шаг
- Рассмотрим эволюцию переносов:
  - разряд i порождает перенос, если $c_i = 0$ и $c_{i+1} = 1$
    - ⋆ тогда $a_i = b_i = 1$, т.е. $a_i \wedge b_i = 1$
  - разряд i сохраняет перенос, если $c_i = 1$ и $c_{i+1} = 1$
    - ⋆ тогда $a_i \vee b_i = 1$

  $\Rightarrow$ $c_i = 1 \Leftrightarrow$ найдется разряд $j < i$ такой, что
    - j порождает перенос, а каждый разряд k, $j < k < i$, сохраняет его
- Положим $p_i = a_i \vee b_i$, $g_i = a_i \wedge b_i$ $\Rightarrow$ $c_i = \bigvee_{j=0}^{i-1} \left(g_j \wedge \bigwedge_{k=j+1}^{i-1} p_k \right)$

★ ADD_n вычисляется за три шага:
шаг 1 — все p_i и g_i; шаг 2 — все c_i; шаг 3 — все s_i

Пример: схема сложения для 4-битных чисел ($n = 4$)