Задачи по булеви функции.

Съдържание

1 Теоретична основа

1.1 Булеви вектори

Както знаем, елементите на $\{0,1\}^n$ са наредените n-орки от нули и единици. Тук ги наричаме "булеви вектори с дължина n" или накратко "n-вектори". За краткост и прегледност ще записваме n-векторите без скоби и запетаи-разделители; примерно, 010110, а не $(0,1,0,1,1,0)$.

Ползваме удобната конвенция имената на векторите да бъдат записвани с удебелени букви (в полиграфията се казва "получерен шрифт", на английски е *boldface*), например $\mathbf{b}$. Ако сме дефинирали някакво име на n-вектор, да кажем $\mathbf{b}$, то неговите елементи именуваме със същото име, само че тях изписваме с нормално дебели букви (*regular face*), например $\mathbf{b} = b_1 b_2 \cdots b_n$, където $b_i \in \{0,1\}$ за $1 \leq i \leq n$.

Нека $x, y \in \{0,1\}$. $x \stackrel{\text{деф}}{=} \overline{y}$ тстк

$$x = \begin{cases} 1, & \text{ако } y = 0 \\ 0, & \text{ако } y = 1 \end{cases}$$

Противоположни вектори са вектори с една и съща дължина, които се различават във всеки елемент; ако $\mathbf{a}$ и $\mathbf{b}$ са n-вектори, те са противоположни тстк

$$\forall i \in \{1, 2, \ldots, n\} : a_i = \overline{b_i}$$

Факта, че $\mathbf{a}$ и $\mathbf{b}$ са противоположни, записваме накратко така: $\mathbf{a} = \overline{\mathbf{b}}$. Добре известно е, че $\overline{\overline{\mathbf{a}}} = \mathbf{a}$, така че $\mathbf{a} = \overline{\mathbf{b}} \leftrightarrow \overline{\mathbf{a}} = \overline{\overline{\mathbf{b}}} = \mathbf{b}$.

Нулев вектор е всеки вектор от една или повече нули $00 \cdots 00$. Ако дължината не е съществена за дискурса, може да казваме членувано *нулеви**ят** вектор*. Нулевият вектор ще записваме с "$\mathbf{0}$". Напълно аналогично, въвеждаме *единичния вектор*, който записваме с "$\mathbf{1}$".

1.2 Дефиниция на "булева функция"

Определение 1. Нека $n \in \mathbb{N}$. *Булева функция с n аргумента* е всяка функция от вида $f : \{0,1\}^n \to \{0,1\}$. *Булева функция* е всяка булева функция с n аргумента, за някое $n \in \mathbb{N}$.

Неформално, булева функция с n аргумента е "раздаване" на нули или единици на n-векторите. Формално, булева функция с n аргумента е множество от 2^n на брой наредени двойки, чиито първи елемент е n-вектор, а вторият елемент е или 0, или 1. Примерно, конюнкцията е булева функция с 2 аргумента, а именно $\{(00, 0), (01, 0), (10, 0), (11, 1)\}$.

Множеството от всички булеви функции с n аргумента ще бележим с " $\mathcal{F}^n$ ". Съгласно изучаваното в раздела Комбинаторика, $|\mathcal{F}^n| = 2^{2^n}$. Примерно, булевите функции с 1 аргумент са 4 на брой, тези с 2 аргумента са 16 на брой, тези с 3 аргумента са 256 на брой, тези с 4 аргумента са $65\,536$ на брой, и така нататък.

Забележете, че n е естествено число, така че може да имаме 0 аргумента, като функциите с 0 аргумента са 2 на брой. Наистина, 0-кратното декартово произведение е $\{()\}$, следователно домейнът е едноелементен и има точно две булеви функции с 0 аргумента, които отъждествяваме с двете булеви константи 0 и 1.

Множеството $\mathcal{F}$ от всички булеви функции е изброимо безкрайно и се бележи с "$\mathcal{F}$". Очевидно

$$\mathcal{F} = \bigcup_{n \in \mathbb{N}} \mathcal{F}^n$$

1.3 Булевите функции като вектори

По дефиниция, всяка булева функция е множество от наредени двойки. Ето пример за булева функция с 3 аргумента:

$$f = \big\{((0,0,0),0), ((0,0,1),1), ((0,1,0),1), ((0,1,1),0)\\ ((1,0,0),1), ((1,0,1),1), ((1,1,0),0), ((1,1,1),1)\big\}$$

Можем да опишем тази функция много по-прегледно с таблица:

			f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Но може да запишем функцията още по-компактно. Оттук насетне приемаме, че n-векторите винаги са подредени отгоре надолу лексикографски, точно както са подредени 3-векторите в тази таблица. Тогава е излишно да ги пишем – знаейки броя на аргументите, ние знаем на кой ред в таблицата, кой е n-векторът.

Ако не пишем векторите, таблицата става със само една колона. В нашия пример:

f
0
1
1
0
1
1
0
1

За да пестим място при писането, записваме функцията не като колона, а хоризонтално:

$$f = 01101101$$

Такова представяне на булева функция ще наричаме *канонично*. Каноничното представяне на булева функция с n аргумента е вектор от 2^n на брой булеви стойности.

Забележете, че дължината на вектора трябва да е точна степен на двойката, за да имаме право да кажем, че този вектор представлява булева функция. И така, ако е даден булев $\mathbf{a}$ вектор с дължина m, то

- ако m е точна степен на двойката, то $\mathbf{a}$ представлява булевата функция f (тя е една единствена) с $\log_2 m$ аргумента, като за всяко $i \in \{0, 1, \ldots, m-1\}$, ако $\mathbf{b}$ е записът на i в двоична позиционна бройна система с дължина $\log_2 m$ и евентуално попълване с нули отпред, то $f(\mathbf{b}) = a_i$;
- ако m не е точна степен на двойката, то $\mathbf{a}$ не представлява запис на булева функция.

Като пример, ако поначало беше даден векторът $\mathbf{a} = 01101101$, веднага виждаме, че дължината му е 8 и това е точна степен на двойката, така че той представлява булевата функция f с 3 аргумента, такава че $f(000) = 0$, $f(001) = 1$, $f(010) = 1$, и така нататък. Въпреки че, строго формално, функцията не е вектор, а множество от наредени двойки, имаме право да я отъждествим с вектора, защото той я определя напълно при споменатите имплицитни допускания.

Като друг пример, векторът 010110 не представлява булева функция, защото дължината му не е точна степен на двойката.

1.4 Аргументи и променливи

Както знаем, "променлива" е количество, което може да приема като стойност кой да е елемент на някакво множество. Неформално, това е кутия, в която можем да сложим кой да е елемент на множеството. Когато говорим за булева функция f с n аргумента, можем да ползваме израза "$f(\mathbf{x})$", където $\mathbf{x}$ е променливата на функцията. Променливата $\mathbf{x}$ взема стойности от множеството $\{0, 1\}^n$. Това множество има структура – то е Декартово произведение. Това ни дава право да ползваме израза "$f(x_1, x_2, \ldots, x_n)$", където x_1, x_2, $\ldots$, x_n са променливи, вземащи стойности от множеството $\{0, 1\}$. Те са две по две различни променливи, въпреки че вземат стойности от едно и също множество. Иначе казано, в Декартовото произведение

$$\underbrace{\{0, 1\} \times \{0, 1\} \times \cdots \times \{0, 1\}}_{n \text{ множители}}$$

асоциираме всеки множител с отделна променлива.

Дотук изглежда, че "променливи на булева функция" и "аргументи на булева функция" са едно и също. Но това не е така. Променливите са имена на аргументите. Като пример, нека пак да разгледаме функцията на 3 аргумента

$$f = 01101101$$

Ако използваме записа "$f(x_1, x_2, x_3)$", ние казваме, че първият аргумент се казва x_1, вторият се казва x_2, а третият се казва x_3. Но можеше да напишем "$f(x_2, x_1, x_3)$". Сега наричаме първия аргумент x_2, втория, x_1, а третия, x_3. Какви имена ще ползваме, няма значение за функцията – те не я променят. Можеше и "$f(x, y, z)$". Имената трябва да са две по две различни.

Ако обаче **вече** сме дефинирали

$$f(x_1, x_2, x_3) = 01101101$$

и **след това** ползваме записа "$f(x_2, x_1, x_3)$", вече говорим за друга функция! Да видим защо. Ако използваме таблица, за да опишем $f(x_1, x_2, x_3)$, таблицата е тази:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Таблица 1: $f(x_1, x_2, x_3)$.

В израза "$f(x_2, x_1, x_3)$", имената "x_1" и "x_2" са разменени по отношение на дефиницията. Таблицата на $f(x_2, x_1, x_3)$ се получава от таблицата на $f(x_1, x_2, x_3)$ чрез размяна на първата и втората колона, а **не** чрез просто преименуване на колоните:

Ясно се вижда, че това е друга функция – каноничното ѝ описание е 01111001, докато началната функция беше 01101101. Забележете, че и в двете таблици, върху един и същи вектор функцията има една и съща стойност; примерно, $f(000) = 0$, $f(100) = 1$, и така нататък. Това е очевидно – стойностите на функцията върху векторите са нейната същностна характеристика. Разменяйки променливите x_1 и x_2 спрямо първоначално дефинираната $f(x_1, x_2, x_3)$, ние в някакъв смисъл променяме наредбата на векторите. Забележете, че в Таблица 2, векторите не са наредени лексикографски отгоре надолу. Променяйки наредбата на векторите, ние пермутираме елементите на колоната f в таблицата, при което може да получим друга функция, както в примера.

x_2	x_1	x_3	$f(x_2, x_1, x_3)$
0	0	0	0
0	0	1	1
1	0	0	1
1	0	1	1
0	1	0	1
0	1	1	0
1	1	0	0
1	1	1	1

Таблица 2: $f(x_2, x_1, x_3)$.

Прочее, свойството на булева функция (а и не само булева; това е в сила за всяка функция на много променливи, които вземат стойности от едно и също множество) да не се променя при пермутация на променливите си, се нарича *симетричност* (Подсекция 1.13).

За променливи, каквито разгледахме досега, казваме, че са *независими*, защото вземат стойностите си независимо една от друга. Да разгледаме възможността някои от тях да са *зависими* и по-точно стойността на едната да е точно равна на стойността на другата. Това се нарича *унификация* на променливи и се отбелязва чрез еднакви имена[†]. Нека пак ползваме като пример

$$f(x_1, x_2, x_3) = 01101101$$

Тогава функцията $f(x_1, x_1, x_2)$ се получава от $f(x_1, x_2, x_3)$ чрез унификация на първите две променливи. Забележете, че $f(x_1, x_1, x_2)$ е функция на **две променливи** и с **три аргумента**. И така, при унификация на променливи, броят на променливите намалява, но броят на аргументите не намалява. Ако мислим за табличното представяне, разглеждаме функцията само върху редовете, в които първата и втората променлива имат една и съща стойност; в някакъв смисъл, зачеркваме останалите редове.

x_1	x_1	x_3	$f(x_1, x_1, x_3)$
0	0	0	0
0	0	1	1
~~0~~	~~1~~	~~0~~	~~1~~
~~0~~	~~1~~	~~1~~	~~0~~
~~1~~	~~0~~	~~0~~	~~1~~
~~1~~	~~0~~	~~1~~	~~1~~
1	1	0	0
1	1	1	1

Таблица 3: $f(x_1, x_1, x_3)$.

Всяка унификация намалява дължината на функцията наполовина. В случая, $f(x_1, x_1, x_2) = 0101$ е с дължина 4. В екстремния вариант, унификация на всички променливи води до

[†]Ако трябва да сме педантични, унификацията е на аргументи, понеже променливите са имената на аргументите.

функция с дължина само 2, независимо от броя на аргументите; това е напълно очаквано, тъй като при унификация на всички променливи остава само една променлива, която има точно две възможни стойности.

Заслужава да се акцентира, че броят на аргументите не намалява при унификация на променливи, така че ако началната функция е с n аргумента, след унификация на променливи функцията пак е с домейн n-вектори, но вече не всички n-вектори, а само тези, които имат едни и същи стойности на дадени позиции. В примера, домейнът на $f(x_1, x_1, x_3)$ е $\{000, 001, 110, 111\}$.

Унификацията на променливи може да бъде описана формално така.

Определение 2. Нека $f(x_1, x_2, \ldots, x_n)$ е булева функция. Нека $1 \leq i < j \leq n$. Нека

$$X = \{\mathbf{a} \in \{0,1\}^n \mid a_i = a_j\}$$

Булевата функция, която се получава от $f(x_1, x_2, \ldots, x_n)$ *чрез унификация на* x_i *и* x_j, е $f\big|_X$.

Записът "$f(x_1, x_2, \ldots, x_n)$", който често ползваме, предполага, че унификация на променливи няма, така че променливите са на брой колкото аргументите, а именно n.

Определение 3. Нека $f(x_1, x_2, \ldots, x_n)$ е булева функция. Нека $1 \leq i \leq n$. Променливата x_i се нарича *фиктивна*, ако

$$f(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, x_{i+2}, \ldots, x_n) = f(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, x_{i+2}, \ldots, x_n)$$

за всяка стойност на $(n-1)$-вектора $x_1 x_2 \cdots x_{i-1} x_{i+1} x_{i+2} \cdots x_n$. Променлива, която не е фиктивна, се нарича *съществена*.

1.5 Булевите функции на един и два аргумента

Както знаем от Подсекция 1.2, $|\mathcal{F}^n| = 2^{2^n}$, така че $|\mathcal{F}^1| = 4$. Следва таблица с четирите булеви функции с един аргумент, написани в колони вдясно от двата 1-вектора. Името на аргумента (променливата) е x.

x	0	$\mathtt{id}(x)$	$\mathtt{neg}(x)$	1
0	0	0	1	1
1	0	1	0	1

Ето разяснения за четирите функции.

- 0 се нарича *константа нула*. В случая е константа нула с един аргумент, но ние ще ползваме "0" за константа нула на произволен брой аргументи.

- $\mathtt{id}(x)$ е идентитет x. Ясно е защо се казва така: нейните стойности съвпадат с тези на x. На практика често тази функция се записва с "x". Въпреки че по този начин ползваме една и съща буква и за променливата, и за функцията, помним, че те са принципно различни неща.

- $\mathtt{neg}(x)$ е отрицание x. Ясно е защо се казва така: нейните стойности съвпадат с тези на $\overline{x}$. На практика често тази функция се записва с "$\overline{x}$".

- 1 се нарича *константа единица*. В случая е константа единица с един аргумент, но, също като при константа нула, ние ще ползваме "1" за константа единица на произволен брой аргументи.

Следва таблица с някои, но не всички, булеви функции с два аргумента. Имената на аргументите (променливите) са x и y.

x	y	$\mathbf{0}$	$x \wedge y$	$\mathtt{id}(x)$	$\mathtt{id}(y)$	$x \oplus y$	$x \vee y$	$x \downarrow y$	$\mathtt{neg}(y)$	$\mathtt{neg}(x)$	$x \to y$	$x \vert y$	$\mathbf{1}$
0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	1	0	0	0	1	1	1	0	0	1	1	1	1
1	0	0	0	1	0	1	1	0	1	0	0	1	1
1	1	0	1	1	1	0	1	0	0	0	1	0	1

Ето разяснения за тези функции, които нямат аналог при един аргумент.

- $x \wedge y$ е конюнкция. Допустимо е да се записва и с амперсанд "$x\&y$" или с конкатенация на променливите "xy".
- $x \oplus y$ е сума по модул две. Това е аналога на изключващото или от съждителната логика. В теорията на булевите функции, сума по модул две има много по-голяма роля, отколкото изключващото или в съждителната логика. Понякога се записва с обикновен плюс "$x + y$", ако няма възможност за двусмислица.
- $\mathtt{neg}(x)$ е отрицание x. Ясно е защо се казва така: нейните стойности съвпадат с тези на $\overline{x}$. На практика често тази функция се записва с "$\overline{x}$".
- $x \vee y$ е дизюнкция. **Не ползвайте** буквата плюс за дизюнкция!
- $x \downarrow y$ се нарича стрелка на Пърс (Peirce). Тази функция е широко известна като NOR, което идва от "negation of or". Недостатък на името "NOR" е, че то предполага композиция на функции; един вид, първо правим OR и после го негираме. Докато буквата "$\downarrow$" подчертава, че става дума за една функция.
- $x \to y$ е импликация. Заслужава да се спомене, че в съждителната логика импликацията има ключово значение, защото е свързана с правенето на изводи, докато в теорията на булевите функции тя няма никакво особено значение.
- $x|y$ се нарича черта на Шефер (Sheffer). Тази функция е широко известна като NAND, което идва от "negation of and". Недостатък на името "NAND" е, че то предполага композиция на функции; един вид, първо правим AND и после го негираме. Докато буквата "$|$" подчертава, че става дума за една функция.

Свойствата на булевите функции на две променливи отговарят точно на свойствата на съответните логически съюзи от съждителната логика. Примерно, конюнкцията е комутативна и асоциативна и дизюнкцията също. Конюнкцията дистрибутира спрямо дизюнкцията и обратното. Негацията на негацията на x е същото като x, негацията на дизюнкцията е конюнкцията на негациите, и така нататък. Заслужават да се споменат няколко ключово важни свойства на сумата по модул две, чиито аналози в съждителната логика не сме разглеждали.

1. $x \oplus y = y \oplus x$. Сумата по модул две е комутатитвна.
2. $x \oplus (y \oplus z) = (x \oplus y) \oplus z$. Сумата по модул две е асоциативна и имаме право да пишем $x \oplus y \oplus z$. Това се обобщава за произволен брой събираеми: записът $x_1 \oplus x_2 \oplus \cdots \oplus x_k$ е недвусмислен; допустимо е накратко да се пише $\bigoplus_{i=1}^{k} x_i$.
3. $x \oplus 0 = x$. Очевидно винаги е вярно, независимо дали x е 0 или 1.
4. $x \oplus 1 = \mathtt{neg}(x)$. Очевидно винаги е вярно, независимо дали x е 0 или 1.

5. $x \oplus x = 0$. Очевидно винаги е вярно, независимо дали x е 0 или 1.

6. $y \oplus x \oplus x = y$. Следва веднага от 5. и 3.

1.6 Композиция

Нека $f(x_1, x_2, \ldots, x_i, \ldots, x_n)$ и $g(y_1, y_2, \ldots, y_m)$ са булеви[†] функции. *Композицията на g на мястото на x_i във f е функцията* $f(x_1, x_2, \ldots, x_{i-1}, g(y_1, y_2, \ldots, y_m), x_{i+1}, x_{i+2}, \ldots, x_n)$. Това е функция, която е **различна** (в общия случай) и от f, и от g. Ако се интересуваме от изчисляването на тази функция, алгоритъм за нейното изчисляване може да се получи от алгоритми за изчисляването на f и на g: алгоритъмът за f вика алгоритъма за g. Новата функция-композиция се дефинира така. Тя е функция h с $n + m - 1$ аргумента и за всеки $(n + m - 1)$-вектор $\mathbf{a}$, стойността ѝ съвпада със стойността на $f(\mathbf{b})$, където

$$
\begin{aligned}
b_1 &= a_1 \\
b_2 &= a_2 \\
&\ldots \\
b_{i-1} &= a_{i-1} \\
b_{i+1} &= a_{i+m} \\
b_{i+2} &= a_{i+m+1} \\
&\ldots \\
b_n &= a_{n+m-1}
\end{aligned}
$$

а b_i има същата стойност като $g(a_i, a_{i+1}, \ldots, a_{i+m-1})$.

Колко са променливите на функцията-композиция? Очевидно множеството от променливите на функцията-композиция е $(\{x_1, \ldots, x_n\} \setminus \{x_i\}) \cup \{y_1, \ldots, y_m\}$. Ако е изпълнено $(\{x_1, \ldots, x_n\} \setminus \{x_i\}) \cap \{y_1, \ldots, y_m\} = \emptyset$[‡], то композицията е функция на $n + m - 1$ променливи. Например, ако са дадени $f(x_1, x_2, x_3, x_4, x_5)$ и $g(x_6, x_7, x_8)$, то композицията $f(x_1, x_2, x_3, g(x_6, x_7, x_8), x_5)$ е функция на $5 + 3 - 1 = 7$ променливи. Възможно е обаче $(\{x_1, \ldots, x_n\} \setminus \{x_i\}) \cap \{y_1, \ldots, y_m\} \neq \emptyset$. В такъв случай броят се получава чрез комбинаторния принцип на включването и изключването: от сумата от мощностите вадим мощността на сечението и вадим още една единица заради x_i. Например, ако са дадени $f(x_1, x_2, x_3, x_4, x_5)$ и $g(x_1, x_2, x_8)$, то композицията $f(x_1, x_2, x_3, g(x_1, x_2, x_8), x_5)$ е функция на $(5 + 3) - 2 - 1 = 5$ променливи.

Това може да се обобщи така. Ако g_1, g_2, …, g_n са булеви функции съответно на m_1, …, m_n променливи, а именно

$$
\begin{aligned}
&g_1(y_{1,1}, \ldots, y_{1,m_1}) \\
&g_2(y_{2,1}, \ldots, y_{2,m_2}) \\
&\ldots \\
&g_n(y_{n,1}, \ldots, y_{n,m_n})
\end{aligned}
$$

[†] Не е необходимо тези f и g да са **булеви** функции, за да може да говорим за композиция. Композиция на функцията g на мястото на x_i във функцията f е мислима дори когато f и g са произволни функции при условие, че кодомейнът на g е същият като i-ия домейн на f. Казано на програмистки жаргон, при условие, че типът на изхода на g е същият като типа на i-ия вход на f.

[‡] С други думи, ако променливите на f без x_i, от една страна, и променливите на g, от друга страна, нямат общи елементи.

то композицията на g_1 на мястото на x_1, на g_2 на мястото на x_2, …, на g_n на мястото на x_n е булевата функция:

$$f(g_1(y_{1,1}, \ldots, y_{1,m_1}), g_2(y_{2,1}, \ldots, y_{2,m_2}), \ldots, g_n(y_{n,1}, \ldots, y_{n,m_n}))$$

1.7 Обобщена конюнкция и обобщена дизюнкция

Нека f_1 е булевата функция конюнкция. Нека x_1, x_2, x_3 и x_4 са булеви променливи. Всеки от следните записи:

$$f_1(x_1, x_2) \qquad f_1(x_1, x_3) \qquad f_1(x_1, x_4) \qquad f_1(x_2, x_3) \qquad f_1(x_2, x_4) \qquad f_1(x_3, x_4)$$

е допустим. От друга страна, следните записи:

$$f_1(x_1, x_2, x_3) \qquad f_1(x_2, x_3, x_4) \qquad f_1(x_1, x_2, x_3, x_4)$$

са, от формална гледна точка, **недопустими**, понеже конюнкцията е функция на точно **две** променливи, а не на повече.

На практика обаче ние говорим за конюнкция на много променливи. Как става това? Ако трябва да сме напълно прецизни, конюнкцията на повече от две променливи не е функцията f_1, а друга функция, която се получава от f_1 чрез подходяща серия от композиции. Като пример да разгледаме следните пет функции:

$$\begin{aligned} & f_1(f_1(f_1(x_1, x_2), x_3), x_4) && f_1(x_1, f_1(x_2, f_1(x_3, x_4))) \\ & f_1(f_1(x_1, x_2), f_1(x_3, x_4)) && \\ & f_1(f_1(x_1, f_1(x_2, x_3)), x_4) && f_1(x_1, f_1(f_1(x_2, x_3), x_4)) \end{aligned} \tag{1}$$

Всяка от тези пет функции е функция на четирите променливи x_1, x_2, x_3 и x_4. Лесно се вижда, че тези функции отговарят биективно на петте[†] начина за скобуване на редицата от променливите. Прочее, тези пет функции са равни—по-прецизно казано, (1) съдържа пет записа на една и съща функция—поради асоциативността на конюнкцията. Лесно се вижда освен това, че и всяка друга линейна наредба на променливите, да кажем $x_2x_4x_1x_3$ би довела до същото – функцията на четири променливи си остава същата; това пък е заради комутативността на конюнкцията.

И така, функцията от (1) е пример за *обобщена конюнкция*. Това ще рече, функция на две или повече променливи, която се получава от обикновената конюнкция чрез серия от композиции. Очевидно, обобщената конюнкция има стойност единица тстк всичките ѝ променливи са единици.

Напълно аналогично дефинираме и *обобщена дизюнкция*: функция на две или повече променливи, която се получава от обикновената дизюнкция чрез серия от композиции; тя има стойност единица тстк поне една от променливите ѝ е единица.

1.8 Други представяния на булевите функции

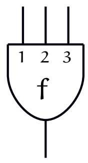
1.8.1 Представяне чрез схеми от функционални елементи

Да допуснем, че ни е дадено устройство, което n входа и точно един изход, за някое $n \in \mathbb{N}$. На входовете се подават булеви стойности. Дадена ни е булева функция f с n аргумента. Казваме, че устройството *реализира функцията* f, ако за всяка комбинация-вектор $\mathbf{x}$ от булеви

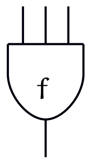
[†]Странична забележка: в комбинаториката, n-тото число на Catalan, което записваме като C_n, е броят на всички начини дадена линейна наредба на $n+1$ елемента, взети от някакво множество A, да бъде скобувана така, че дадена бинарна операция над A да бъде приложена над въпросната линейна наредба. В конкретния пример, множеството A е $\{0, 1\}$, броят на обектите е 4, бинарната операция е f_1, а линейната наредба е $x_1x_2x_3x_4$. Действително, $C_3 = 5$, което точно отговаря на броя на функциите в (1).

стойности на входовете, на изхода "излиза" булева стойност $f(\mathbf{x})$. Стандартно допускане е, че входовете на устройството са именувани; тоест всеки вход си има идентичност. Всяко такова устройство наричаме *функционален елемент*[†]. За целите на този курс ние разглеждаме само идеализирани функционални елементи, но такива функционални елементи може да бъдат реализирани физически, което е в основата на цифровата схемотехника.

Ще изобразяваме функционалните елементи примерно така:



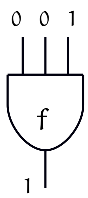
Елементът се рисува с нещо като полукръг, като входовете (в случая те са три на брой и са номерирани с 1, 2 и 3) са откъм правата част, а изходът, който винаги е точно един, е от заоблената част. Името на функцията е написано върху елемента. Естествено, функцията трябва да е такава, че броят на аргументите ѝ да е точно равен на броя на входовете на елемента. Ако се разберем, че входовете са номерирани отляво надясно, можем спокойно да изпускаме изписването на номерата им; в такъв случай същият функционален елемент би бил нарисуван така:



Нека отново f е следната функция:

$$f = 01101101$$

Тогава върху входния вектор 001, f има стойност 1. Това изобразяваме върху функционалния елемент ето така:

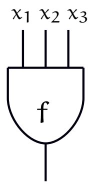


Очевидно допускаме, че има посока на "движение на информацията", която е винаги от входовете към изхода.

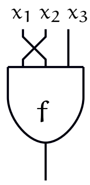
Схемите от функционални елементи са над някакво множество от булеви променливи. Удобно е да си представяме, че променливите "текат" по "жиците"[‡], които влизат във входовете на функционалните елементи. Например, ако искаме да изобразим функционален елемент, съответстващ на $f(x_1, x_2, x_3)$, правим такава диаграма:

[†]На английски терминът е *gate*.

[‡] Реалните функционални елементи са електронни устройства, по чиито жици текат токове и има напрежения, като напреженията се **интерпретират** като нули или единици. Идеализираните функционални

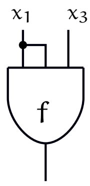


Забележете, че следното свързване на входовете на функционалния елемент към трите променливи е **съществено различно** от предишното



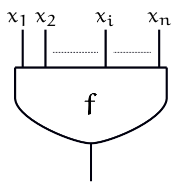
по същите причини, поради които $f(x_1, x_2, x_3)$ в общия случай е различна функция от $f(x_2, x_1, x_3)$ (припомнете си разликата между Таблица 1 и Таблица 2). И така, схемите от функционални елементи чудесно онагледяват какво става при пермутация на променливите в записа на функцията.

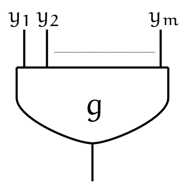
Схемите от функционални елементи чудесно онагледяват и понятието унификация на променливи. Ако разгледаме същия функционален елемент, който ползвахме в примерите горе, то унификацията на x_1 с x_2 (припомнете си Таблица 3) изглежда така върху елемента:



Ясно се вижда, че след унификацията, променливите са само две. Черната точка е точка на разклонение на сигнала, при което идващата отгоре стойност x_1 се размножава и "влиза" в двата входа вляво. В терминологията на електротехниката, първите два входа на елемента са "на късо".

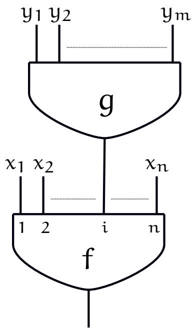
Използвайки каскадно свързани функционални елементи можем много елегантно да илюстрираме композицията на функция на мястото на променлива на друга функция. Както в Подсекция 1.6, нека $f(x_1, x_2, \ldots, x_i, \ldots, x_n)$ и $g(y_1, y_2, \ldots, y_m)$ са булеви функции. Съответните им функционални елементи си представяме така:





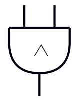
елементи, които разглеждаме тук, са абстракции и при тях за токове и напрежения не става дума. Върху техните "жици" "текат" нули и единици. При реалните функционални елементи имат някакви закъснения—в природата нищо не се случва мигновено—така че при всяка промяна на входовете е необходимо някакво време, типично от порядъка на наносекунди или пикосекунди при модерните върхови цифрови електронни схеми. Идеализираните функционални елементи, които разглеждаме тук, нямат закъснения и при тях сигналите се разпространяват неограничено бързо.

Тогава композицията на g на мястото на x_i във f се описва, в термините на функционалните елементи, като свързване на изхода на елемента на g към i-ия вход на елемента на f:

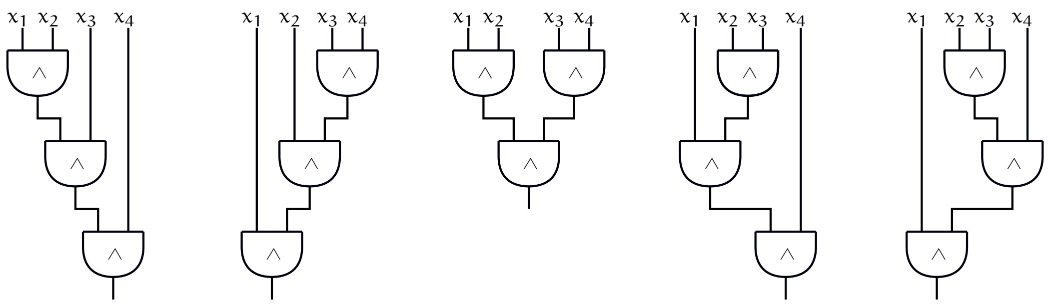


i-ият вход продължава да съществува, но вече не е именуван с "x_i", защото променлива x_i вече няма в смисъл, че не се ползва. Активните променливи, с други думи, тези, които се ползват, са x_1, ..., x_{i-1}, x_{i+1}, ..., x_n, y_1, ..., y_m. Тази каскада от функционални елементи реализира функцията $f(x_1, x_2, \ldots, x_{i-1}, g(y_1, y_2, \ldots, y_m), x_{i+1}, x_{i+2}, \ldots, x_n)$.

Функциите обобщена конюнкция и обобщена дизюнкция, за които стана дума в Подсекция 1.7, може да се илюстрират чрез каскадни свързвания на функционални елементи тип конюнкция или дизюнкция. Нека ето този функционален елемент реализира булевата функция конюнкция:



Тогава петте композиции от (1) могат да бъдат представени така:



Както се каза в Подсекция 1.7, поради асоциативността на конюнкцията, петте композиции от (1) са всъщност пет начина да бъде написана една и съща функция. Следователно, петте различни свързвания на функционални елементи от последната фигура реализират една и съща функция, а именно обобщена конюнкция на четири променливи.

1.8.2 Представяне чрез хиперкуб

Понякога е много удобно да мислим за булевите функции на n променливи в термините на *хиперкуб*. n-мерен хиперкуб е обобщение на редицата от геометрични обекти `точка`, `отсечка`, `квадрат`, `куб` и така нататък:

- точката е 0-мерен хиперкуб, който е атомарен в смисъл, че няма структура,
- отсечката е 1-мерен хиперкуб, състоящ се от 2 точки и едномерния обект, който ги свързва—можем да кажем, в някакъв смисъл, "който е ограден от тях",
- квадратът е 2-мерен хиперкуб, състоящ се от 4 точки, 4 отсечки и двумерния обект, ограден от тях,
- кубът е 3-мерен хиперкуб, състоящ се от 8 точки, 12 околни ръба, 6 квадрата и тримерния обект, ограден от тях,
- и така нататък.

От тази гледна точка, n-мерният хиперкуб е общият член на тази редица. Той е геометричен обект в n-мерното пространство. Може да мислим за хиперкуба като за обект, състоящ се от:

- 2^n точки, които са върховете му,
- $n2^{n-1}$ отсечки, които са околните му ръбове,
- $\binom{n}{2}2^{n-2}$ квадрата, които са околните му стени,
- $\binom{n}{3}2^{n-3}$ куба,
- и така нататък
- $\binom{n}{n-1}2^{n-(n-1)} = 2n$ на брой, $(n-1)$-мерни обекта,
- един n-мерен обект, ограден от $(n-1)$-мерните обекти.

Лесно се вижда, че k-мерните компоненти на n-мерния хиперкуб са $\binom{n}{k}2^{n-k}$ на брой.

Може да пренебрегнем геометричния аспект на хиперкуба и да мислим за него като за чисто комбинаторен обект по следния начин:

- Върховете му са векторите от $\{0,1\}^n$. Това са 0-мерните компоненти на n-мерния хиперкуб.
- Два вектора са *съседни* тогава и само тогава, когато се различават в точно една позиция. Например, ако $n = 3$, векторите 001 и 011 се различават в точно една позиция (втората) и те задават един околен ръб на 3-мерния хиперкуб. И така, всеки околен ръб се идентифицира с двата върха, които му притадлежат, а те се различават в точно една позиция. С други думи, 1-мерните компоненти са всички (ненаредени) двойки вектори, които се различават в точно една позиция.
- Аналогично, всяка околна стена се идентифицира с четирите върха, които ѝ принадлежат. С други думи, 2-мерните компоненти са всички (ненаредени) четворки вектори, такива че има точно две позиции, в които тези четири вектора се различават.

- Аналогично, 3-мерните компоненти са всички (ненаредени) осморки вектори, такива че има точно три позиции, в които се различават.
- И така нататък.
- n-мерната компонента е точно една: това е множеството от всички, 2^n на брой, n-вектори.

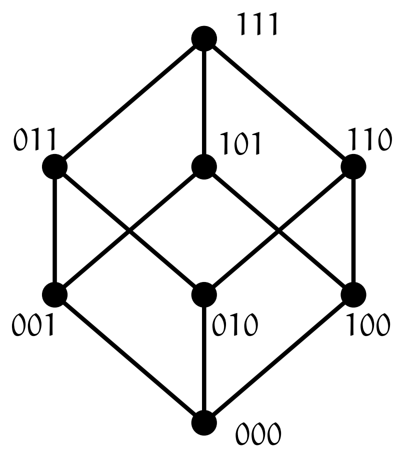
Тогава общият брой компоненти на n-мерния хиперкуб е:

$$\sum_{k=0}^{n} \binom{n}{k} 2^{n-k} = \sum_{k=0}^{n} \binom{n}{n-k} 2^{n-k} = \sum_{0 \le k \le n} \binom{n}{n-k} 2^{n-k} =$$
$$\sum_{0 \ge -k \ge -n} \binom{n}{n-k} 2^{n-k} = \sum_{n \ge n-k \ge 0} \binom{n}{n-k} 2^{n-k} = \sum_{0 \le k \le n} \binom{n}{k} 2^{k} = \sum_{0 \le k \le n} \binom{n}{k} 2^{k} 1^{n-k} = 3^n$$

Като пример да разгледаме 3-мерния хиперкуб, записан **напълно подробно**:

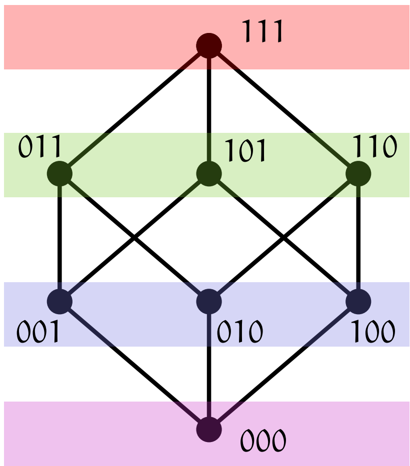
$$
\begin{aligned}
&\{\\
&\ \{\{000\},\{001\},\{010\},\{011\},\{100\},\{101\},\{110\},\{111\}\}, \quad //8 \text{ върха}\\
&\ \{\{000,100\},\{000,010\},\{010,011\},\{001,011\},\{000,100\},\{001,101\},\\
&\quad \{011,111\},\{010,110\},\{100,101\},\{101,111\},\{111,110\},\{110,100\}\} \quad //12 \text{ ръба}\\
&\ \{\{000,001,011,010\},\{000,100,101,001\},\{001,101,111,011\},\\
&\quad \{010,110,100,000\},\{010,110,111,011\},\{100,101,111,110\}\}, \quad //6 \text{ стени}\\
&\ \{\{000,001,010,011,100,101,110,111\}\} \quad //1 \text{ обем}\\
&\}
\end{aligned}
$$

Обикновено хиперкубът се рисува като граф: само върховете и страните. Това означава, че 0-мерните и 1-мерните компоненти се изобразяват, а останалите, не. Така е много по-прегледно. Но ние знаем, че хиперкубът като комбинаторен обект е съвкупност от обектите от всички размерности, от 0 до n †. Ето типично изображение на 3-мерния хиперкуб:

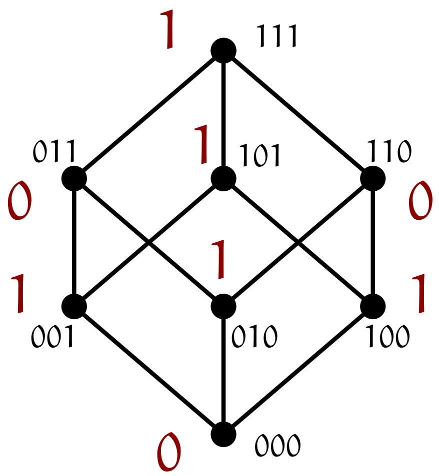


† Ако говорим за *граф-хиперкуб*, тогава имаме предвид обекта, който е съвкупност само от 0-мерните компоненти (върховете) и 1-мерните компоненти, които в този контекст наричаме "ребра". n-мерен граф-хиперкуб не е същото нещо като n-мерен хиперкуб: графът-хиперкуб е подмножество на хиперкуба. Хиперкубът съдържа и компонентите от размерности ≥ 2.

Върховете на n-мерния хиперкуб се разбиват на $n+1$ *слоя*. Върховете от един слой са точно тези вектори, които имат един и същи брой единици. Броят на единиците може да е $0, 1, \ldots, n$, затова и слоевете са $n+1$. Когато говорим за слой k, $0 \le k \le n$, имаме предвид слоя от векторите, всеки от които има точно k единици. Очевидно слой k има $\binom{n}{k}$ вехктора в себе си. Ето същия хиперкуб, като четирите слоя са указани с различни цветове:



Всяка булева функция на n променливи може да бъде разглеждана като асоцииране на всеки връх на n-мерния хиперкуб (помним, че върховете му са точно n-векторите) с една булева стойност. Например, функцията $f = 01101101$ от миналата подсекция се изобразява върху хиперкуба така:



Стойностите на функцията върху векторите са написани с червено.

От казаното дотук изглежда, че каноничното представяне на дадена булева функция и представянето чрез хиперкуб са едно и също нещо. Всъщност, разлика има и тя е само в наредбата на векторите. При каноничното представяне, векторите са наредени лексикографски[†], а при представянето с хиперкуб те са наредени от частичната наредба $\preccurlyeq \subseteq \{0,1\}^n \times \{0,1\}^n$, дефинирана по следния начин:

$$\forall \mathbf{a}, \mathbf{b} \in \{0,1\}^n : \mathbf{a} \preccurlyeq \mathbf{b} \leftrightarrow (\forall i \in \{1,2,\ldots,n\} : a_i \leq b_i) \qquad (2)$$

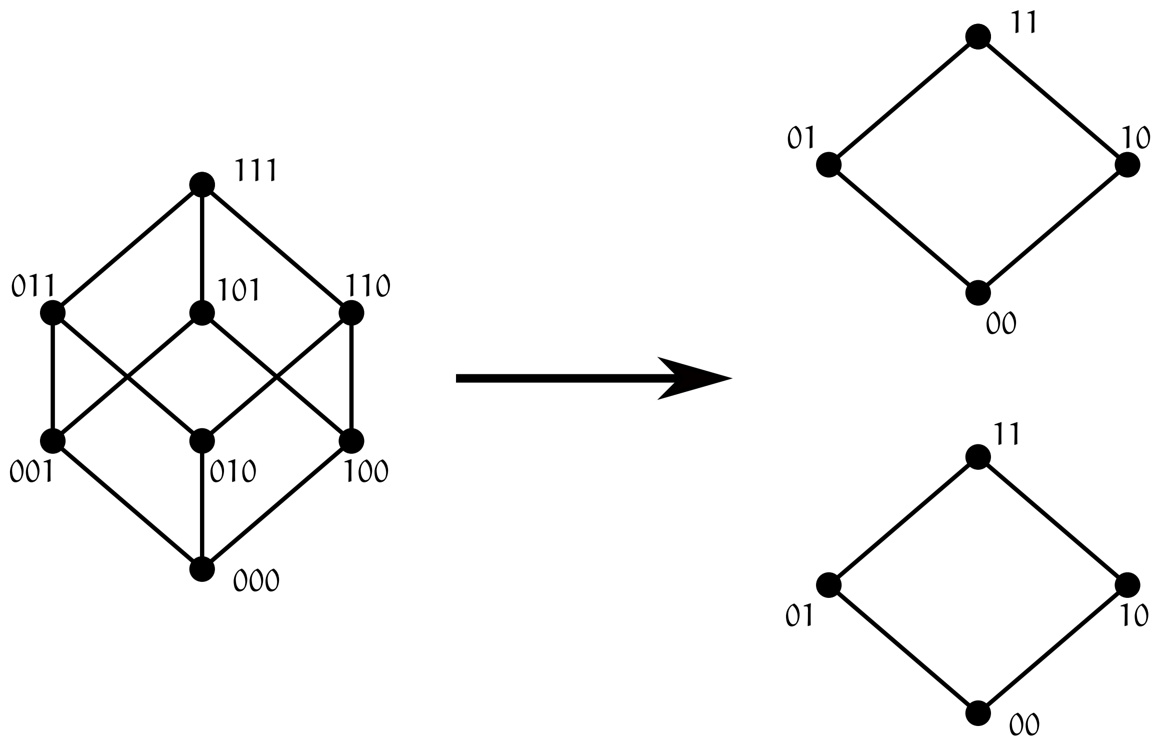
Това е частична наредба, която не е линейна, понеже има двойки вектори, които не са сравними (спрямо нея), например 011 и 100. Тази частична наредба е полезна за осмислянето на различни понятия и решаването на много задачи от областта на булевите функции. На горния пример всъщност е показана диаграмата на Hasse на частичната наредба $\preccurlyeq$ върху 3-векторите.

Както казахме вече, съседни вектори са такива, които се различават в точно една позиция[‡]. Съседството на вектори може да осмислим и в термините на хиперкуба: два негови вектора са съседни, ако са в съседни слоеве. Съседство на вектори може да осмислим и чрез релацията $\preccurlyeq$ (виж (2)). А именно, ако $\mathbf{a}$ и $\mathbf{b}$ са n-вектори, то те са съседни тогава и само тогава, когато $\mathbf{a} \prec \mathbf{b}$ или $\mathbf{b} \prec \mathbf{a}$, където релацията $\prec$ се дефинира така:

$$\forall \mathbf{a}, \mathbf{b} \in \{0,1\}^n : \mathbf{a} \prec \mathbf{b} \leftrightarrow \mathbf{a} \preccurlyeq \mathbf{b} \wedge \mathbf{a} \neq \mathbf{b} \wedge \neg\exists \mathbf{c}(\mathbf{a} \preccurlyeq \mathbf{c} \preccurlyeq \mathbf{b}) \qquad (3)$$

Например, $0010 \prec 0011$ и $0010 \prec 1010$, но $0010 \not\prec 1111$, въпреки че $0010 \preccurlyeq 1111$.

Срязване на n*-мерния хиперкуб в* i*-тата димензия* е понятие, което първо ще онагледим с пример. Ето срязване на 3-мерния хиперкуб във втората димензия:



За удобство, нека да мислим за хиперкуба като за граф-хиперкуб, тоест съвкупност от върхове и ребра. Срязването се състои в премахване на i-тата позиция на всички вектори-върхове, след което тяхната дължина става $n-1$, и премахването на точно тези ребра, които

[†]Лексикографската наредба е линейна.

[‡]Освен това, за да говорим за съседство на вектори, трябва те да имат една и съща дължина. При вектори с различни дължини за съседство не може да става дума.

са от вида:

$$\{\alpha 0\beta, \alpha 1\beta\}$$

където α е булев вектор с дължина $i-1$, а β е булев вектор с дължина $n-i$. В примера със срязването на 3-мерния хиперкуб във втората димензия, ребрата, които махаме, са точно

$$\{000, 010\}, \{001, 011\}, \{100, 110\}, \{101, 111\}$$

Ако гледаме на хиперкуба не като на граф, а като на "истински" хиперкуб с компоненти от всички възможни размерности, ясно е, че срязването води до изчезването на n-мерната компонента, както и до намаляването на броя на k-мерните компоненти от $\binom{n}{k}2^{n-k}$ на $\binom{n-1}{k}2^{n-k-1}$, тъй като резултатът от срязването е появата на два нови хиперкуба, всеки с размерност $n-1$.

От казаното досега може да не е ясно защо настояваме да се казва, че срязваме именно в i-тата размерност. Например, на последната фигура от един куб се получават два квадрата и по нищо не личи точно в коя от трите размерности е бил срязан куба. Отговорът на тази забележка е, че хиперкубът ни интересува в контекста на булевите функции, когато върховете му са "маркирани" с нули или единици – стойностите на булевата функция. При срязването асоциацията между върхове и стойности на функцията **се запазва**, така че в общия случай резултатът от срязването в различни размерности е **различен**.

1.8.3 Представяне чрез формули

Формула е чисто синтактично понятие. Средношколското разбиране за "формула" е "прост алгоритъм", например "формулата за лицето на кръг с радиус r е $S = \pi r^2$". Тук ние възприемаме съвсем друго разбиране за "формула". Формула е всеки стринг, конструиран над дадена азбука съгласно дадени правила. Етимологията на думата е следната: на латински "formula" е умалително от "forma". Не е грешка да казваме "форма" вместо "формула".

Формулите на булевите функции може да се дефинират индуктивно по следния начин. Да фиксираме изброимо безкрайно множество от булеви променливи $\{x_0, x_1, \ldots\}$. Нека Σ е азбуката:

$$\Sigma = \{f, x, 0, 1, \ldots, 9, (,), ,\}$$

С червено са записани буквите от езика, който ще опишем, а именно езика от формулите на булевите функции, а с черно са буквите от метаезика, който използваме, за да опишем езика от формулите на булевите функции. Нека $\Sigma_d = \{0, 1, \ldots, 9\}$. Нека $\widetilde{\Sigma}$ е множеството от стрингове над Σ_d, които са валидни записи[†] на числа в десетична позиционна бройна система. Нека ι е стандартната десетична позиционна бройна система, тоест биекцията

$$\iota : \widetilde{\Sigma} \to \mathbb{N}$$

която знаем от училище. Нека *enum* е произволно изброяване на всички булеви функции, тоест биекция $enum : \mathcal{F} \to \mathbb{N}$. За удобство вземаме най-простото и естествено изброяване на булевите функции:

- за всяко $n \in \mathbb{N}$, ако $f \in \mathcal{F}^n$ и $g \in \mathcal{F}^{n+1}$, то $enum(f) < enum(g)$,
- за всяко $n \in \mathbb{N}$, ако $f, g \in \mathcal{F}^n$ и $f \neq g$, то $enum(f) < enum(g)$ тогава и само тогава, когато каноничното представяне на f предхожда лексикографски каноничното представяне на g.

[†]Например 0017 не е валиден запис заради двете водещи нули.

Нека изброените от *enum* булеви функции са f_0, f_1 и така нататък. Да разгледаме началото на изброяването. f_0 е константа нула с нула аргумента, тоест 0. f_1 е константа единица с нула аргумента, тоест 1. f_2 е константа нула с една аргумент, тоест 00. f_3, f_4 и f_5 са съответно 01, 10 и 11. f_6 е константа нула с два аргумента, тоест 0000. И така нататък. f_{21} е константа единица с два аргумента, тоест 1111. f_{22} е константа нула с три аргумента, тоест 00000000. И така нататък. f_{277} е константа единица с три аргумента, тоест 11111111. f_{278} е константа нула с четири аргумента, тоест 00000000. И така нататък. f_{65813} е константа единица с четири аргумента, тоест 1111111111111111. f_{65813} е константа нула с пет аргумента. И така нататък.

Тогава дефинираме "формула на булева функция" чрез следната индуктивна дефиниция. Тя формализира понятието "формула на булена функция" и въвежда дълбочина на формула, която бележим с ν.

Определение 4 (Формули на булевите функции, дълбочина на формула)**.** Всеки стринг $\phi \in \Sigma^+$ е *формула на булева функция* тогава и само тогава, когато в сила е точно едно от двете:

- **База.** $\phi = \mathtt{x}\alpha$, където $\alpha \in \widetilde{\Sigma}$. $\nu(\phi) \stackrel{\text{деф}}{=} 0$.
- **Индуктивна стъпка.** $\phi = \mathtt{f}\alpha\mathtt{(}\phi_1\mathtt{,}\phi_2\mathtt{,}\ldots\mathtt{,}\phi_n\mathtt{)}$ където ϕ_1, ϕ_2, ..., ϕ_n са формули на булеви функции, $\alpha \in \widetilde{\Sigma}$ и $enum^{-1}(\iota(\alpha))$ има точно n аргумента[†]. Последното е важно! Забележете, че $\mathtt{f999999(x1,x2)}$ е синтактично невалиден запис, защото булевата функция номер деветстотин деведетдесет и девет хиляди деветстотин деветдесет и девет е с пет аргумента съгласно номерацията *enum*.

$$\nu(\phi) \stackrel{\text{деф}}{=} \max\{\nu(\phi_1), \nu(\phi_2), \ldots, \nu(\phi_n)\} + 1$$

Нека Φ е множеството от всички формули на булеви функции. Функцията $\nu : \Phi \to \mathbb{N}$, която току-що дефинирахме, се нарича *дълбочина*. $\square$

Неформално, ако си представим формулата, реализирана чрез каскада от функционални елементи, то дълбочината на формулата е максималният брой функционални елементи, през които трябва да премине сигналът.

Дотук сме дефинирали, чисто синтактично, формулите на булевите функции. Не сме казали нищо за техния смисъл, или, иначе казано, за тяхната *семантика*. Семантиката можем да дефинираме, използвайки индуктивната дефиниция на синтаксиса, като аналогът на синтактичната операция "вмъкване на стринг на мястото на подстринг" (има се предвид вмъкването на формулите ϕ_i на местата на имената на променливите) е семантичната "композиция на функция на мястото на променлива в друга функция".

Определение 5 (Семантика на формулите на булевите функции)**.** В контекста на Определение 4:

- семантиката на всяка формула с дълбочина нула $\mathtt{x}\alpha$ е булевата променлива $x_{\iota(\alpha)}$.
- семантиката на всяка формула с дълбочина единица $\mathtt{f}\alpha\mathtt{(x}\beta_1\mathtt{,x}\beta_2\mathtt{,}\ldots\mathtt{,x}\beta_n\mathtt{)}$, където $\beta_i \in \widetilde{\Sigma}$ за $1 \leq i \leq n$, е булевата функция $f_i(x_{k_1}, x_{k_2}, \ldots, x_{k_n})$, където $f_i = enum^{-1}(\iota(\alpha))$ и x_{k_j} за $1 \leq j \leq n$ е булевата променлива, чийто индекс k_j е $\iota(\beta_j)$.

[†]Забележете, че $\iota(\alpha)$ е число, а $enum^{-1}(\iota(\alpha))$ е една от всички булеви функции, защото $enum^{-1} : \mathbb{N} \to \mathcal{F}$.

- семантиката на всяка формула с дълбочина повече от единица $\phi = f\alpha(\phi_1,\phi_2,\ldots,\phi_n)$ е композицията на семантиките на ϕ_1, …, ϕ_n на местата на съответно първата, …, n-тата променлива на функцията f_i, където $f_i = enum^{-1}(\iota(\alpha))$.

Лесно се вижда, че при изредените правила булевата функция, която е семантиката на някаква формула, е **една единствена**, но обратното не е вярно: за всяка функция има **безброй много** формули, на които тя е семантика. Ако булевата функция f е семантиката на формулата ϕ ще казваме, че f *съответства на* ϕ, или че ϕ *реализира* f. И така, ϕ реализира точно една функция f, но безброй много формули съответстват на функцията f.

Често срещана задача в Компютърните науки е: дадени са две формули, да се реши дали са еквивалентни. Тоест, дали съответната им булева функция е една и съща, или не. Това е частен случай на общата задача: дадени са два синтактични обекта (някакви стрингове, изградени по някакви правила), да се определи дали семантиката им е една и съща, или не, като семантиката е някаква добре дефинирана функция.

Формулите, които ще разглеждаме на практика, по правило са само над ограничено множество булеви функции, което най-често е крайно. Нека $F \subseteq \mathcal{F}$ е непразно; дали е крайно или не, няма значение. *Формулите над* F е това подмножество на Φ, чиито елементи-формули реализират функциите от F; с други думи, в индуктивната стъпка на Определение 4, стрингът от цифри α е такъв, че $f_{\iota(\alpha)}$ е елемент на F.

Функциите, чиито формули ще използваме в този курс, са "стандартните" булеви функции на две променливи: конюнкцията, дизюнкцията, импликацията, сумата по модул 2, стрелката на Peirce и чертата на Sheffer, а така също и отрицанието, което е функция на една променлива. Има смисъл множеството от функциите, чиито формули ще се ползват, да бъде пълно множество (вж. Секция 1.12).

1.9 Дизюнктивна Нормална Форма и Съвършена Дизюнктивна Нормална Форма

Определение 4 и Определение 5 дават мощен механизъм за изграждане на формули, чиито смисъл (семантика) са булеви функции. Този механизъм има съществен недостатък: формулите, изградени по него, са практически нечетими. Заради това, в Подсекция 1.9 и Подсекция 1.10 ще въведем формули, които се изграждат по съвсем различни правила. Тези формули са лесно четими и са универсални в смисъл, че всяка булева функция се реализира от някои от тези формули (Теорема 1). Що се отнася до семантиката обаче, тези формули обаче не са над произволно множество от булеви функции, а само над отрицание, конюнкция и дизюнкция.

Нека са фиксирани краен брой булеви променливи x_1, …, x_n за $n \geq 1$. *Литерал* ще наричаме всяко име на променлива или всяко име на променлива с черта отгоре (отрицание). Литералите от първия вид се наричаме *положителни*, а от втория – *отрицателни*. Веднага подчертаваме, че литералите са **букви** и като такива са качествено различни от самите променливи, понеже буквите са понятия от синтактичното ниво, а променливите са от по-високото семантично ниво. Това, че използваме един и същи запис "x_1" и за името на променлива (което е буква), и за самата променлива, не води до объркване, защото опитният читател винаги може да разбере от контекста дали става дума за синтактичното ниво или за семантичното ниво. Примери за положителни литерали са x_1, x_4 и така нататък. Примери за отрицателни литерали са $\overline{x_1}$, $\overline{x_3}$ и така нататък. Отново: това са букви. Ерго, ако променливите са n на брой, азбуката на формулите (които ще строим) съдържа $2n$ букви само заради променливите; а именно, n на брой положителни литерали и n на брой отрицателни литерали.

Конюнктивна клауза е всяка непразна формула, която се състои от конкатенация на литерали, такива че всяко име на променлива се появява най-много веднъж – било като положителен, било като отрицателен литерал. Ако променливите са $x_1, \ldots, x_6$, примери за конюнктивни клаузи са $x_1x_3x_4$, $x_1\overline{x_4}$, $x_1x_2x_3x_4x_5x_6$, $\overline{x_3}$, $x_2\overline{x_5}\,\overline{x_6}$ и така нататък. Не е задължително, но е силно препоръчително имената да се записват отляво надясно в нарастващ ред на индексите.

Пълна конюнктивна клауза е конюнктивна клауза, която съдържа точно n литерала. С други думи, това е непразна формула, която се състои от конкатенация на литерали, такива че всяко име на променлива се появява точно веднъж – било като положителен, било като отрицателен литерал. Ако променливите са $x_1, \ldots, x_6$, примери за пълни конюнктивни клаузи са $x_1x_2x_3x_4x_5x_6$, $x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4}\,\overline{x_5}\,x_6$ и така нататък. Не е задължително, но е силно препоръчително имената да се записват отляво надясно в нарастващ ред на индексите.

Дизюнктивна нормална форма, съкратено ДНФ, е формула, която се състои от една или повече различни[†] конюнктивни клаузи, конкатенирани с буквата "$\vee$". В горния контекст, примери за ДНФ са $x_2\overline{x_3}x_4$, $x_1x_4 \vee x_2\overline{x_5}\,\overline{x_6} \vee \overline{x_2}\,\overline{x_3}$ и така нататък. *Съвършена дизюнктивна нормална форма*, съкратено *СъвДНФ* е дизюнктивна нормална форма, в която участват само пълни конюнктивни клаузи. В горния контекст, пример за СъвДНФ е $x_1\overline{x_2}x_3x_4x_5x_6 \vee \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}\,\overline{x_5}\,\overline{x_6}$.

Семантиката на литералите, конюнктивните клаузи и ДНФ е очевидната:

- Семантиката на всеки положителен литерал x_i е функцията идентитет-x_i. Семантиката на всеки отрицателен литерал $\overline{x_i}$ е функцията отрицание-на-x_i.
- Семантиката на всяка конюнктивна клауза $\lambda_1\lambda_2\cdots\lambda_k$, където λ_j са литерали за $1 \leq j \leq k$, е композицията $f_{\mathrm{con}}(f_1, f_2, \ldots, f_k)$, където f_{con} е обобщената конюнкция на k променливи, а f_j е семантиката на λ_j, за $1 \leq j \leq k$.
- Семантиката на всяка ДНФ $\phi_1 \vee \phi_2 \vee \cdots \vee \phi_k$, където ϕ_j е конюнктивна клауза за $1 \leq j \leq k$, е $f_{\mathrm{dis}}(f_1, f_2, \ldots, f_k)$, където f_{dis} е обобщената дизюнкция на k променливи, а f_j е семантиката на ϕ_j, за $1 \leq j \leq k$.

Пример за СъвДНФ. Като пример да разгледаме формулата (тя е СъвДНФ, ако $n = 6$)

$$\phi = x_1\overline{x_2}x_3x_4x_5x_6 \vee \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}\,\overline{x_5}\,\overline{x_6}$$

Очевидно, нейната семантика е булевата функция—да я наречем h—на шестте променливи x_1, $\ldots$, x_6, която има стойност 1 върху векторите 000000 и 101111 и има стойност 0 върху всички останали, 62 на брой, вектори. Сега да си представим, че трябва да запишем h чрез формула, изградена съгласно индуктивното Определение 4. Естествено, има безброй начини да сторим това, но нека се опитаме да напишем формула, която е аналогична на ϕ. Лесно се вижда, че ни трябват формули за функциите отрицание, конюнкция и дизюнкция. Функцията h е равна на някаква композиция от тези функции[‡], а именно на дизюнкция от някакви конюнкции. Аналогът на това в синтактичния свят на формулите е: че формула за h може да бъде

[†]Кога две конюнктивни клаузи са различни? Ако настояваме индексите на променливите да са в нарастващ ред отляво надясно, то две конюнктивни клаузи са различни тстк са различни като стрингове. Без това ограничение, можем да дефинираме "различни формули" и в частност различни конюнктивни клаузи чрез релация на еквивалентност.

[‡]Тази композиция не е единствена заради комутативността на дизюнкцията и конюнкцията.

получена, като във формула за дизюнкция заместим стринговете-имена на променливи с някакви формули за конюнкции.

Да направим формула за h точно по Определение 4, като използваме червен цвят за буквите и́. Функциите отрицание, конюнкция и дизюнкция имат номера съответно 4, 7 и 13 в изброяването *enum*, тоест, това са съответно f_4, f_7 и f_{13}. Ето пример за формула, съответна на h:

$$\begin{aligned}\psi = {} & f13(f7(x1, f7(f4(x2), f7(x3, f7(x4, f7(x5, x6))))),\\ & f7(f4(x1), f7(f4(x2), f7(f4(x3), f7(f4(x4), f7(f4(x5), f4(x6)))))))\end{aligned}$$

Очевидно ϕ е несравнимо по-лека за четене от ψ, макар че са еквивалентни, имайки една и съща семантика. Може да възникне въпросът, защо изобщо ползваме тромавата конструкция на Определение 4, след като има начин да се записват еквивалентни формули, които са много по-лесни за четене. Отговорът е, че конструкцията на Определение 4 има чисто теоретично значение. Там искахме да дефинираме прецизно и кратко "формула" и "функция, съответна на формула", а не сме имали за цел получените формули да са кратки и ясни. Говорейки за ДНФ искаме друго – кратък и много ясен запис на формулите. За тази цел е много по-удачно да се въведат литерали и конюнктивни клаузи и чрез тях и буквите "$\vee$" да се дефинират ДНФ.

Доказателството на теоремата на Boole се основава на факта, че всяка булева функция с поне един аргумент, която не е константа-нула, има една единствена СъвДНФ.

1.10 Конюнктивна Нормална Форма и Съвършена Конюнктивна Нормална Форма

Нека са фиксирани краен брой булеви променливи $x_1, \ldots, x_n$ за $n \geq 1$. *Дизюнктивна клауза* е всяка непразна формула, която се състои от литерали, конкатенирани с буквата "$\vee$", такива че всяко име на променлива се появява най-много веднъж – било като положителен, било като отрицателен литерал. Ако променливите са $x_1, \ldots, x_6$, примери за дизюнктивни клаузи са:

$$\begin{aligned}& x_1 \vee x_3 \vee x_4,\\ & x_1 \vee \overline{x_4},\\ & x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6,\\ & \overline{x_3},\\ & x_2 \vee \overline{x_5} \vee \overline{x_6}\end{aligned}$$

и така нататък. Не е задължително, но е силно препоръчително имената да се записват отляво надясно в нарастващ ред на индексите.

Пълна дизюнктивна клауза е дизюнктивна клауза, която съдържа точно n литерала. С други думи, това е непразна формула, която се състои от литерали, конкатенирани с буквата "$\vee$", такива че всяко име на променлива се появява точно веднъж – било като положителен, било като отрицателен литерал. Ако променливите са $x_1, \ldots, x_6$, примери за пълни дизюнктивни клаузи са:

$$\begin{aligned}& x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6,\\ & x_1 \vee \overline{x_2} \vee \overline{x_3} \vee \overline{x_4} \vee \overline{x_5} \vee x_6\end{aligned}$$

и така нататък. Не е задължително, но е силно препоръчително имената да се записват отляво надясно в нарастващ ред на индексите.

Конюнкивна нормална форма, съкратено КНФ, е формула, която се състои от конкатенация на една или повече различни дизюнктивни клаузи, като, ако дизюнктивните клаузи са повече от една, всяка от тях е оградена от чифт скоби; ако дизюнктивната клауза е само една, скоби не се ползват. В горния контекст, примери за КНФ са:

$$x_2 \vee \overline{x_3} \vee x_4,$$
$$(x_1 \vee x_4 \vee x_2 \vee \overline{x_5})(\overline{x_6} \vee \overline{x_2} \vee \overline{x_3})$$

и така нататък.

Съвършена конюнктивна нормална форма, съкратено СъвКНФ е конюнктивна нормална форма, в която участват само пълни дизюнктивни клаузи. В горния контекст, пример за СъвКНФ е $(x_1 \vee \overline{x_2} \vee x_3 \vee x_4 \vee x_5 \vee x_6)(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3} \vee \overline{x_4} \vee \overline{x_5} \vee \overline{x_6})$.

Семантиката на КНФ е очевидната:

- Семантиката на всеки положителен литерал x_i е функцията идентитет-x_i. Семантиката на всеки отрицателен литерал $\overline{x_i}$ е функцията отрицание-на-x_i.
- Семантиката на всяка дизюнктивна клауза $\lambda_1 \vee \lambda_2 \vee \cdots \vee \lambda_k$, където λ_j са литерали за $1 \leq j \leq k$, е композицията $f_{\text{dis}}(f_1, f_2, \ldots, f_k)$, където f_{dis} е обобщената дизюнкция на k променливи, а f_j е семантиката на λ_j, за $1 \leq j \leq k$.
- Семантиката на всяка КНФ $(\phi_1)(\phi_2)\cdots(\phi_k)$, където ϕ_j е дизюнктивна клауза за $1 \leq j \leq k$, е $f_{\text{con}}(f_1, f_2, \ldots, f_k)$, където f_{con} е обобщената конюнкция на k променливи, а f_j е семантиката на ϕ_j, за $1 \leq j \leq k$.

1.11 Валюация на ДНФ или КНФ

При ДНФ и КНФ променливите са първични: първо е дадено множество от булеви променливи, от него по някакви правила правим формули и булевите функции се появяват като семантики на тези формули. Това е радикално различен подход от подхода в Подсекция 1.2, в който функциите са първични, а променливите са някакви именувания на аргументите. При ДНФ и КНФ не говорим за унификации на променливи: ако някаква ДНФ или КНФ ползва повече от веднъж литерал (положителен или отрицателен) на една и съща променлива, в някакъв смисъл унификация има, но това не си заслужава да се споменава, защото е очевидно. Докато при подхода в Подсекция 1.2 има смисъл да се говори експлицитно за унификация, ако има такава.

Нека е дадена ДНФ или КНФ ϕ над някакви булеви променливи $x_1, \ldots, x_n$. Нека ϕ реализира булевата функция $f(x_1, x_2, \ldots, x_n)$. Всеки вектор $\mathbf{b} \in \{0,1\}^n$ представлява "раздаване" на конкретни стойности на променливите. Такова раздаване на конкретни стойности на променливите се нарича *булева оценка*, или *валюация* (на английски често е *truth assignment*). Очевидно валюациите са 2^n на брой. Можем да мислим, че всяка валюация е функция $t : \{x_1, \ldots, x_n\} \to \{0,1\}$.

Ако α е литерал във ϕ и x_i е променливата на α, то *стойността на* α *под* t е:

- $t\big|_{x_i}$, ако α е положителен литерал (стойността на литерала съвпада със стойността, която t дава на променливата x_i);
- $\overline{t\big|_{x_i}}$, ако α е отрицателен литерал (стойността на литерала е обратната на стойността, която t дава на променливата x_i).

Ако λ е клауза във ϕ, то стойността на λ под t се определя по правилата на конюнкцията, ако λ е конюнктивна клауза (ϕ е ДНФ), или на дизюнкцията, ако λ е дизюнктивна клауза (ϕ е КНФ).

Стойността на цялата ϕ под t се определя по правилата на дизюнкцията, ако ϕ е ДНФ, или на конюнкцията, ако ϕ е КНФ.

1.12 Пълнота на множества от булеви функции

Нека $F \subseteq \mathcal{F}$. Неформално, множеството F е *пълно*, ако всяка булева функция $f \in \mathcal{F}$ може да бъде представена чрез функциите от F. Това наистина е неформално, защото какво означава "представяне"?

Формално, нека $[F]$ означава затварянето на F спрямо композиция, като затварянето $[F]$ може да се дефинира чрез следната индуктивна дефиниция.

Определение 6 (Затваряне на множество от булеви функции)**.** Нека $F \subseteq \mathcal{F}$. Тогава $[F]$ е най-малкото по включване множество, такова че $[F] \subseteq \mathcal{F}$ и

- $[F]$ съдържа всички функции от F.
- Нека f е функция, която се съдържа в $[F]$. Нека f има поне един аргумент. Нека идентифицираме някои от променливите на f. Получената функция се съдържа в $[F]$.
- Нека f и g са произволни функции от $[F]$. Нека f има $n \geq 1$ аргумента. Нека $i \in \{1, \ldots, n\}$. Тогава композицията на g на мястото на i-тата променлива на f също се съдържа в $[F]$.

Определение 7 (Пълнота на множество от булеви функции)**.** В контекста на Определение 6, F е пълно множество, ако $[F] = \mathcal{F}$. □

Теорема 1 (теорема на Boole)**.** Множеството от трите булеви функции конюнкция, дизюнкция и отрицание е пълно.

Доказателство: Нека $F = \{\wedge, \vee, \text{neg}\}$. Ще докажем, че всяка булева функция f има формула над F. От това следва веднага, че $[F] = \mathcal{F}$.

Първо да допуснем, че $f \neq 0$. Ще докажем, че съществува СъвДНФ ϕ, такава че f е семантиката на ϕ. Нека f има n аргумента. Тогава f, в каноничното си представяне, е вектор с дължина 2^n. Щом f не е константа нула, този вектор има поне една единица. Нека f има точно k единици, където $1 \leq k \leq 2^n$. Нека $\mathbf{b}_1$, $\mathbf{b}_2$, …, $\mathbf{b}_k$ са булевите вектори от $\{0,1\}^n$, такива че

$$
\begin{aligned}
f(\mathbf{b}_1) &= 1 \\
f(\mathbf{b}_2) &= 1 \\
&\cdots \\
f(\mathbf{b}_k) &= 1
\end{aligned}
$$

Нека x_1, x_2, …, x_n са булеви променливи, над които ще построим ϕ. Построяваме точно k пълни конюнктивни клаузи λ_1, λ_2, …, λ_k по следните правила. Нека $b_{i,j}$, където $1 \leq i \leq k$ и $1 \leq j \leq n$, е j-ият елемент на вектора $\mathbf{b}_i$. За всяко $i \in \{1, \ldots, k\}$ и $j \in \{1, \ldots, n\}$:

- ако $b_{i,j} = 0$, то променливата x_j участва в λ_i с отрицателния си литерал $\overline{x_i}$.
- ако $b_{i,j} = 1$, то променливата x_j участва в λ_i с положителния си литерал x_i.

Тогава $\phi = \lambda_1 \vee \lambda_2 \vee \cdots \vee \lambda_k$.

Ще докажем, че f е семантиката на ϕ. Наистина,

- за всеки вектор $\mathbf{a} \in \{0, 1\}^n$, такъв че $f(\mathbf{a}) = 0$ (тоест, $\mathbf{a}$ не е нито един от $\mathbf{b}_1$, $\mathbf{b}_2$, ..., $\mathbf{b}_k$), за всяко $i \in \{1, \ldots, k\}$ е вярно, че поне един литерал в клаузата λ_i имат стойност 0. Тогава, по правилата на конюнкцията, клаузата λ_i има стойност 0. Тогава, по правилата на дизюнкцията, цялата формула ϕ има стойност 0.
- за всеки вектор $\mathbf{a} \in \{0, 1\}^n$, такъв че $f(\mathbf{a}) = 1$ (тоест, $\mathbf{a}$ е един от $\mathbf{b}_1$, $\mathbf{b}_2$, ..., $\mathbf{b}_k$), за някое $i \in \{1, \ldots, k\}$ е вярно, че всеки литерал в клаузата λ_i имат стойност 1. Тогава, по правилата на конюнкцията, клаузата λ_i има стойност 1. Тогава, по правилата на дизюнкцията, цялата формула ϕ има стойност 1.

Заключаваме, че наистина е семантиката на ϕ.

Остава да разгледаме случая, в който $f = 0$. Следната формула

$$x_1 \, \overline{x_1}$$

реализира функция константа нула. Тази формула не е ДНФ съгласно формалните правила, защото не е конюнктивна клауза, съдържайки два литерала на една и съща променлива. Но е очевидно, че реализира константа нула и че е формула над множеството от булеви функции конюнкция и отрицание. И това е краят на доказателството на теоремата на Boole. □

Ето пример за прилагането на конструкцията на СъвДНФ от теоремата на Boole. Нека $f(x_1, x_2, x_3)$ е отново тази функция:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Функцията има стойност 1 върху точно пет вектора. На тези вектори съответстват следните клаузи:

- На вектора 001 съотвества $\overline{x_1}\,\overline{x_2}\,x_3$.
- На вектора 010 съотвества $\overline{x_1}\,x_2\,\overline{x_3}$.
- На вектора 100 съотвества $x_1\,\overline{x_2}\,\overline{x_3}$.
- На вектора 101 съотвества $x_1\,\overline{x_2}\,x_3$.
- На вектора 111 съотвества $x_1\,x_2\,x_3$.

Конструкцията построява тази СъвДНФ:

$$\overline{x_1}\,\overline{x_2}\,x_3 \vee \overline{x_1}\,x_2\,\overline{x_3} \vee x_1\,\overline{x_2}\,\overline{x_3} \vee x_1\,\overline{x_2}\,x_3 \vee x_1\,x_2\,x_3$$

Нека читателят се убеди сам(а), че семантиката на тази формула е точно f.

И едно предупреждение. Конструкцията на съответна формула-СъвДНФ може да е много неефикасна. В току-що разгледания пример може да опростим така:

$$\begin{aligned}
&\overline{x_1}\,\overline{x_2}\,x_3 \vee \overline{x_1}\,x_2\,\overline{x_3} \vee x_1\,\overline{x_2}\,\overline{x_3} \vee x_1\,\overline{x_2}\,x_3 \vee x_1\,x_2\,x_3 = \\
&\overline{x_2}\,x_3\,(\overline{x_1} \vee x_1) \vee \overline{x_1}\,x_2\,\overline{x_3} \vee x_1\,\overline{x_2}\,\overline{x_3} \vee x_1\,x_2\,x_3 = \\
&\overline{x_2}\,x_3 \vee \overline{x_1}\,x_2\,\overline{x_3} \vee x_1\,\overline{x_2}\,\overline{x_3} \vee x_1\,x_2\,x_3
\end{aligned}$$

В екстремалния пример, когато $f = \mathbf{1}$, конструкцията генерира СъвДНФ с $n2^n$ литерала; а именно, всички пълни конюнктивни клаузи, конкатенирани чрез "$\vee$". А константа единица може да се получи от формулата $x_1 \vee \overline{x_1}$.

Задачата за генериране на формула с минимална дължина, съответна на дадена функция, в общия случай е практически нерешима. Но въпреки това конструкцията на СъвДНФ е полезна – тя е средство, което ни позволява да конструираме механизъм (СъфДНФ), който реализира някакво поведение (функцията). Известни са алгоритми, чрез които можем значително да намалим дължината на формулата, запазвайки семантиката; въпреки че в общия случай е практически невъзможно да генерираме еквивалентна формула с гарантирано минимална дължина, в много случаи е възможно да съкратим формулата драстично.

И още нещо. Всяка булева функция, която е различна от константа единица, има СъвКНФ, която се конструира по правила, които са дуални на правилата за конструиране на СъвКНФ от теоремата на Boole. Правилата са тези: разглеждаме само векторите, върху които функцията има стойност 0, за всеки такъв вектор $\mathbf{b}_i$ добавяме точно една пълна дизюнктивна клауза λ_i:

- ако $b_{i,j} = 0$, то променливата x_j участва в λ_i с положителния си литерал x_i.
- ако $b_{i,j} = 1$, то променливата x_j участва в λ_i с отрицателния си литерал $\overline{x_i}$.

Тогава $\phi = (\lambda_1)(\lambda_2) \cdots \vee (\lambda_k)$.

Като пример, нека пак $f(x_1, x_2, x_3)$ е тази функция:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Съответната СъвКНФ е $(x_1 \vee x_2 \vee x_3)(x_1 \vee \overline{x_2} \vee \overline{x_3})(\overline{x_1} \vee \overline{x_2} \vee x_3)$.

Теорема 2. Нека $F, G \subseteq \mathcal{F}$. Нека F е пълно. Нека $\forall f \in F : f \in [G]$. Тогава G е пълно.

Доказателство: Иска се да покажем, че $[G] = \mathcal{F}$. Това, че $[G] \subseteq \mathcal{F}$ е очевидно, защото всеки елемент на $[G]$ е булева функция. Остава да покажем, че $\mathcal{F} \subseteq [G]$.

Ще покажем, че $\forall h \in \mathcal{F} : h \in [G]$. Нека h е произволна булева функция. По условие, F е пълно множество, така че $h \in [F]$. Това означава, че h се получава като композиция на функции от F и може би унификации на променливи. По условие, всяка функция от F се получава като композиция на функции от G и може би унификации на променливи. Тогава очевидно h се получава като композиция на функции от G и може би унификации на променливи. □

Следствие 1. Множеството от булеви функции $\{\wedge, \text{neg}\}$ е пълно.

Доказателство: Ще използваме Теорема 1 и Теорема 2. От Теорема 1 знаем, че множеството $F = \{\wedge, \vee, \text{neg}\}$ е пълно. От Теорема 2 знаем, че ако представим всяка от функциите на F чрез елементите на G, имаме доказателство за пълнотата на G. Но $F \cap G = \{\wedge, \text{neg}\}$, така че единствено за дизюнкцията трябва да покажем, че може да се "получи" от конюнкция и отрицание. Наистина,

$$
\begin{aligned}
& x \vee y = && // \text{ двойно отрицание} \\
& \overline{\overline{x \vee y}} = && // \text{ De Morgan} \\
& \overline{\overline{x} \wedge \overline{y}}
\end{aligned}
$$

□

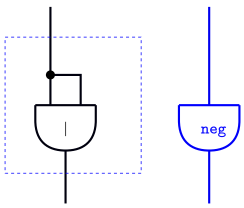
Следствие 2. Множеството от булеви функции $\{\,|\,\}$ е пълно.

Доказателство: Ще използваме Теорема 1 и Следствие 1. Ще изразим и конюкцията, и отрицанието чрез чертата на Sheffer. Наистина,

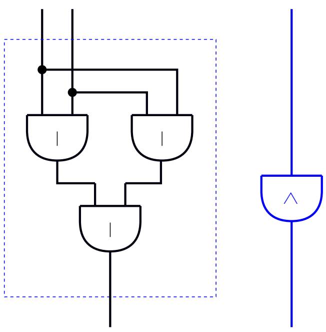
$$\overline{x} = x \,|\, x \tag{4}$$

$$x \wedge y = \overline{x \,|\, y} = (x \,|\, y) \,|\, (x \,|\, y) \tag{5}$$

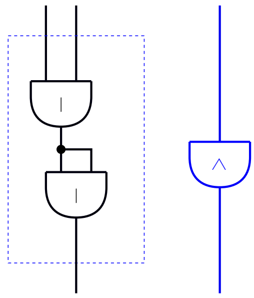
(4), изразено чрез функционален елемент тип черта на Sheffer, изглежда така:



(5), изразено чрез функционален елемент тип черта на Sheffer, изглежда така:



Конюнкция може да се получи чрез функционален елемент тип черта на Sheffer и с по-проста схема:



Забележете, че формулата (5) не може да бъде опростена (съкратена) по аналогичен начин, защото формулите не предоставят механизъм за "пренасяне чрез жица" на вече построена функция, какъвто механизъм е напълно естествен за схемите. Това е илюстрация на факта, че схемите може да са доста по-компактни от съответните формули заради възможността за пренос чрез жица на нещо вече построено (*reusability*). □

1.13 Разни

Функция (може дори да не е булева, но типовете на променливите трябва да са едни и същи) на n променливи, да я наречем $f(x_1, x_2, \ldots, x_n)$ се нарича *симетрична*, ако стойността ѝ се запазва при всяка пермутация на променливите. С други думи,

$$f(x_1, x_2, \ldots, x_n) = f(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$$

за всяка пермутация $\pi(1), \pi(2), \ldots, \pi(n)$ на вектора $1, 2, \ldots, n$. Например, ако $n = 3$, то:

$$f(x_1, x_2, x_3) = f(x_1, x_3, x_2) = f(x_2, x_1, x_3) = f(x_2, x_3, x_1) = f(x_3, x_1, x_2) = f(x_3, x_2, x_1)$$

Ако пък $n = 2$, то:

$$f(x_1, x_2) = f(x_2, x_1)$$

Очевидно комутативността е частен случай на симетричността.

2 Задачи

Задача 1. Намерете $|\mathcal{F}^n|$.

Решение: Добре известно е, че броят на тоталните функции с краен домейн X и краен кодомейн Y е $|Y|^{|X|}$. Прилагаме тази формула с $|X| = 2^n$ и $|Y| = 2$ и получаваме $|\mathcal{F}^n| = 2^{2^n}$. □

Задача 2. Нека

$$S = \left\{ f \in \mathcal{F}^n \,|\, \forall \boldsymbol{w}, \boldsymbol{z} \in \{0,1\}^n : \boldsymbol{w} = \overline{\boldsymbol{z}} \to f(\boldsymbol{w}) = \overline{f(\boldsymbol{z})} \right\}$$

Намерете $|S|$ (като функция на n, разбира се).

Решение: С други думи, търси се броят на булевите функции, които върху противоположни вектори имат противоположни стойности[†].

Да групираме n-векторите по двойки $\mathbf{a}, \mathbf{b}$, такива че $\mathbf{a} = \overline{\mathbf{b}}$. За всяка функция $f \in X$ и за всяка от тези двойки е изпълнено следното. Стойността на функцията върху единия елемент от двойката се определя от стойността на функцията върху другия елемент от двойката. Иначе казано, стойностите, които функцията има върху *половината* от n-векторите, я определят напълно.

Но n-векторите са 2^n, така че половината от тях са $\frac{1}{2}2^n = 2^{n-1}$ на брой. Имайки предвид това, виждаме, че броят на въпросните функции е равен на броя на всички булеви функции върху $n-1$ променливи. Отговорът е $|S| = 2^{2^{n-1}}$. □

Задача 3. Нека

$$X = \left\{ f \in \mathcal{F}^n \mid \forall \boldsymbol{w}, \boldsymbol{z} \in \{0,1\}^n : \boldsymbol{w} = \overline{\boldsymbol{z}} \rightarrow f(\boldsymbol{w}) = f(\boldsymbol{z}) \right\}$$

Намерете $|X|$.

Решение: В тази задача се търси броят на функциите, които имат една и съща стойност върху противоположни вектори. Отговорът е $|X| = 2^{2^{n-1}}$ със практически същите съображения като в решението на Задача 2. □

Задача 4. Намерете броя на булевите функции на n променливи, които имат стойност 1 върху точно k вектора.

Решение: Отговорът очевидно е $\binom{2^n}{k}$. □

Задача 5. Да се намери броят на симетричните булеви функции на n променливи.

Решение: Съгласно определението на симетрична функция, става дума за булеви функции, които запазват стойността си при произволно разместване на стойностите на елементите на входния вектор. Входният вектор се състои от нули и единици, следователно се иска върху всички входни вектори **с един и същи брой единици**, стойността на функцията да е една и съща. С други думи, иска се върху всеки слой на хиперкуба функцията да има една и съща стойност. Слоевете на n-мерния хиперкуб са $n+1$, върху всеки от тях стойността е една и съща, а стойностите върху различни слоеве са произволни една спрямо друга. Тогава отговорът е 2^{n+1}. □

Задача 6. Кои са симетричните булеви функции на 2 променливи?

Решение: С други думи, кои са комутативните функции, тъй като при $n = 2$, свойството симетричност съвпада със свойството комутативност. Съгласно Задача 5, тези функции са

[†] Такива функции се наричат *самодвойствени* (на английски *self-dual*). Просто *двойствена* функция на $f \in \mathcal{F}^n$ е единствената $g \in \mathcal{F}^n$, за която е изпълнено $\forall \mathbf{a} \in \{0,1\}^n : g(\mathbf{a}) = \overline{f(\overline{\mathbf{a}})}$. Самодвойствените функции са тези, които са двойствени на себе си.

$2^{2+1} = 8$ на брой. В каноничното представяне, това са функциите

$$
\begin{aligned}
f_0 &= 0000 \\
f_1 &= 0001 \\
f_6 &= 0110 \\
f_7 &= 0111 \\
f_8 &= 1000 \\
f_9 &= 1001 \\
f_{14} &= 1110 \\
f_{15} &= 1111
\end{aligned}
$$

$\square$

Задача 7. Да се определи броят на булевите функции на n променливи за $n \geq 2$, които запазват стойността си при размяна на променливите x_1 и x_2.

Решение: С други думи, иска се

$$f(x_1, x_2, x_3, \dots, x_n) = f(x_2, x_1, x_3, \dots, x_n)$$

за всички възможни стойности на $x_1, \dots, x_n$. Решението ще получим след като съобразим колко различни входни вектори има по отношение на тази задача.

Лесно се вижда, че ако $x_1 = x_2 = 0$ или $x_1 = x_2 = 1$, разместването на x_1 и x_2 няма значение и функцията запазва стойността си по очевидни причини при разместването на x_1 и x_2. Да разбием множеството от всички входни вектори на 4 равномощни подмножества съгласно четирите възможности за подвектора x_1x_2:

$$
\begin{aligned}
A &= 0 \times 0 \times \underbrace{\{0,1\} \times \cdots \times \{0,1\}}_{n-2 \text{ пъти}} \\
B &= 0 \times 1 \times \underbrace{\{0,1\} \times \cdots \times \{0,1\}}_{n-2 \text{ пъти}} \\
C &= 1 \times 0 \times \underbrace{\{0,1\} \times \cdots \times \{0,1\}}_{n-2 \text{ пъти}} \\
D &= 1 \times 1 \times \underbrace{\{0,1\} \times \cdots \times \{0,1\}}_{n-2 \text{ пъти}}
\end{aligned}
$$

Всяко от тези множества има мощност 2^{n-2}. Да разгледаме булевите функции на n променливи без ограничения. Всяка булева функция на n променливи без ограничения има 2^{n-2} стойности върху векторите от A, върху векторите от B, върху векторите от C и върху векторите от D. Всички тези стойности на функцията, на брой $4 \times 2^{n-2} = 2^n$ може да са единици или нули независимо една от друга, откъдето и броят на всички функции е $= 2^{2^n}$.

Ако имаме предвид ограничението от условието на задачата, виждаме, че върху всеки вектор от C, стойността на функцията трябва да съвпада със стойността ѝ върху точно един вектор от B. От друга страна, върху векторите от A стойностите на функцията са произволни – без значение какви стойности има върху останалите три множества. Аналогично, върху векторите от D стойностите на функцията са произволни – без значение какви стойности има върху останалите три множества.

За да получим броя на търсените функции, достатъчно е да игнорираме едно от множествата B и C, да кажем, че игнорираме C, и да разглеждаме само A, B и D. За всеки вектор $\boldsymbol{z} \in A \cup B \cup D$, стойността на функцията е независима от стойността ѝ върху кой да е друг вектор $\boldsymbol{z}' \in A \cup B \cup D$. Тогава отговорът е

$$2^{2^{|A|+|B|+|D|}} = 2^{3 \times 2^{n-2}}$$

Ако заместим $n = 2$, получаваме 2^3, което точно съвпада с отговора на Задача 6. □

Задача 8. Да се намери броят на булевите функции на n променливи, които приемат стойност 1 върху поне една двойка противоположни входни вектори.

Решение: Отговорът очевидно е $= 2^{2^n} - |A|$, където A е множеството от булевите функции на n променливи, които не приемат стойност 1 върху никои два противоположни вектора. С други думи, за всеки входен вектор $\boldsymbol{a}$ и за всяка функция $f \in A$ е изпълнено:

$$(f(\boldsymbol{a}) = 0 \wedge f(\overline{\boldsymbol{a}}) = 0) \vee \tag{6}$$
$$(f(\boldsymbol{a}) = 0 \wedge f(\overline{\boldsymbol{a}}) = 1) \vee \tag{7}$$
$$(f(\boldsymbol{a}) = 1 \wedge f(\overline{\boldsymbol{a}}) = 0) \tag{8}$$

Двойките противоположни вектори на брой са $\frac{1}{2}2^n = 2^{n-1}$. За всяка такава двойка показахме, че възможностите са точно 3 (за да може функцията да бъде от A). И така, $|A| = 3^{2^{n-1}}$. Отговорът тогава е

$$2^{2^n} - 3^{2^{n-1}} \tag{9}$$

Тук може да възникне следното питане. Щом $2^{2^n} - 3^{2^{n-1}}$ е мощността на непразно множество, то очевидно това е положително число за всяко n. Ако обаче не знаем комбинаторните съображения, довели до отговора, а просто видим $2^{2^n} - 3^{2^{n-1}}$, може да се запитаме, това не може ли да е отрицателно за някакви n. С помощта на математическия анализ можем да докажем, че не може да е отрицателно при неограничено нарастване на n:

$$\lim_{n\to\infty} \frac{2^{2^n}}{3^{2^{n-1}}} = \lim_{n\to\infty} \frac{2^{2^n}}{2^{(\log_2 3)2^{n-1}}} = \lim_{n\to\infty} 2^{(2^n - (\log_2 3)2^{n-1})} = \lim_{n\to\infty} 2^{(2-\log_2 3)2^{n-1}} = \infty$$

понеже $2 > \log_2 3$.

Следното доказателство на същото твърдение е на Добромир Кралчев:

$$2^{2^n} - 3^{2^{n-1}} = 2^{2\times 2^{n-1}} - 3^{2^{n-1}} = 4^{2^{n-1}} - 3^{2^{n-1}} > 0$$

Тази задача може да се реши и с други съображения. Нека B е подмножеството на $\mathcal{F}^n$ от функциите, които приемат стойност 1 върху поне една двойка противоположни вектори. Ние търсим $|B|$. Нека B_k за $1 \leq k \leq 2^{n-1}$ е подмножеството от тези функции, които приемат стойност 1 върху точно k двойки противоположни вектори. Очевидно B се разбива на B_1, …, $B_{2^{n-1}}$, така че

$$|B| = \sum_{k=1}^{2^{n-1}} |B_k|$$

Лесно се вижда, че

$$|B_k| = \binom{2^{n-1}}{k} 3^{(2^{n-1}-k)}$$

Съображенията са следните: по $\binom{2^{n-1}}{k}$ начина можем да изберем от всички 2^{n-1} двойки противоположни вектори такива, върху които функцията да има стойност 1, а за всяка от останалите $2^{k-1} - k$ двойки вектори имаме точно 3 възможности (виж (6)), така щото функцията да няма стойност 1 върху двата елемента на двойката. И така, отговорът е:

$$\sum_{k=1}^{2^{n-1}} \binom{2^{n-1}}{k} 3^{(2^{n-1}-k)} \tag{10}$$

От (9) и (10) извеждаме (с комбинаторни разсъждения) тъждеството:

$$2^{2^n} - 3^{2^{n-1}} = \sum_{k=1}^{2^{n-1}} \binom{2^{n-1}}{k} 3^{(2^{n-1}-k)}$$

□

Задача 9. За колко булеви функции f на n променливи е изпълнено следното: ако $f(\mathbf{a})$ има стойност 1, то f има стойност 1 върху всеки вектор, който има поне толкова единици, колкото $\mathbf{a}$?

Решение: С други думи, ако f има стойност 1 върху някакъв вектор $\mathbf{a}$, то f има стойност 1 върху всички вектори от слоя на хиперкуба, към който слой принадлежи $\mathbf{a}$, и освен това f има стойност 1 върху всички вектори от всички следващи слоеве. Веднага се вижда, че всяка такава функция има една и съща стойност върху всеки слой на хиперкуба и се определя еднозначно от това, къде е "границата" между нулите и единиците върху хиперкуба;. По-подробно казано, върху някакви последователни слоеве на хиперкуба (може и да няма такива), започвайки от слой 0, функцията има стойност само 0, и после върху всички останали слоеве (може и да няма такива), функцията има стойност само 1. Тъй като слоевете са $n+1$, то има точно $n+2$ такива функции. □

Задача 10. За колко булеви функции f на n променливи е изпълнено следното: ако $f(\mathbf{a})$ има стойност 1, то f има стойност 1 върху всеки вектор, който има повече единици от $\mathbf{a}$?

Решение: Задачата е подобна на Задача 9, но само донякъде. В Задача 10:

- или има някакъв "граничен слой" в хиперкуба, нека да е слой k, в който за първи път се появява стойност на функцията 1, като върху всички слоеве с по-малък номер функцията е задължително 0, а върху всички слоеве с по-голям номер функцията е задължително 1,

- или такъв граничен слой няма, тоест изобщо няма единици, тоест функцията е константа-нула[†].

[†]Благодарности на Добромир Кралчев за посочването на това!

Първо да сметнем колко са функциите от търсения вид, които имат поне една единица (с други думи, не са константа-нула). Числото k (номерът на граничния слой) може да е най-малко 0 и най-много n. Тъй като в слоеве с номера $0, \ldots, k-1$ и $k+1, \ldots, n$ нещата са фиксирани, единственото, което варира, е как точно са "раздадени" нулите и единиците в слой k по такъв начин, че да има поне една единица.

Знаем, че слой k има мощност $\binom{n}{k}$. Всички начини да "раздадем" нули и единици на неговите елементи са $2^{\binom{n}{k}}$ на брой. От това число вадим единица, за да отразим факта, че върху поне един вектор от слой k функцията е единица (с други думи, изваждаме от разглеждането раздаването само на нули). И така, за дадено k, броят на начините функцията да има поне една единица върху векторите на слой k е $2^{\binom{n}{k}} - 1$. А отговорът, съгласно принципа на разбиването, е:

$$\sum_{k=0}^{n} \left(2^{\binom{n}{k}} - 1\right) = \sum_{k=0}^{n} \left(2^{\binom{n}{k}}\right) - n - 1 \tag{11}$$

Към (11) добавяме единица, защото функцията може да е константа-нула, и получаваме

$$\sum_{k=0}^{n} \left(2^{\binom{n}{k}}\right) - n$$

□

Задача 11. Да се намери броят на булевите функции на n променливи, които нямат фиктивни променливи.

Решение: Нека $\mathbf{G}^n$ е множеството от булевите функции на n променливи, които нямат фиктивна променлива. Нека $\mathbf{H}^n$ е множеството на булевите функции на n променливи, които имат поне една фиктивна променлива. Очевидно $|\mathbf{G}^n| + |\mathbf{H}^n| = 2^{2^n}$, съгласно комбинаторния принцип на разбиването, така че

$$|\mathbf{G}^n| = 2^{2^n} - |\mathbf{H}^n|$$

Ако променливите са $x_1, \ldots, x_n$, нека $\mathbf{H}^n(i_1, \ldots, i_k)$ означава множеството от булевите функции, в които променливи $x_{i_1}, \ldots, x_{i_k}$ са гарантирано фиктивни (може да има и други фиктивни, но тези със сигурност са фиктивни), където $k \in \{1, \ldots, n\}$ и $1 \leq i_1 < i_2 < \cdots < i_k \leq n$.

Да разгледаме без ограничение на общността променливата x_1. В колко функции тя е фиктивна? Може да има и други фиктивни променливи, може и да няма – пита се, за колко функции е изпълнено x_1 да е фиктивна? От дефиницията на фиктивна променлива имаме изискването за всяка от тези функции, да кажем f, да е изпълнено:

$$f(0, x_2, x_3, \ldots, x_n) = f(1, x_2, x_3, \ldots, x_n)$$

Ако си представим входните вектори, наредени лексикографски отгоре надолу, иска се функцията (която е колона с височина 2^n) да е такава, че горната половина на колоната да е точно като долната половина. Ето пример за функция на три променливи, в която x_1 е фиктивна:

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Виждаме, че за да бъде x_1 фиктивна, стойността на функцията върху половината вектори определя стойността ѝ върху другата половина. Броят на функциите, в които x_1 е фиктивна, е $2^{2^{n-1}}$. Същото е в сила и за всяка друга променлива, но за другите променливи половините вектори не са горната и долната; за x_2, например, колоната от стойности върху първата и третата четвъртини трябва да е същата като колоната от стойности върху втората и четвъртата четвъртини, и така нататък.

Всяка променлива на функцията може да е фиктивна. Но $|\mathbf{H}^n|$ **не е** $n \times 2^{2^{n-1}}$, тъй като този израз брои някои функции по няколко пъти. Забележете, че има функции с повече от една фиктивна променлива. Ето пример за функция, в която и x_1, и x_2 са фиктивни:

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

От $n \times 2^{2^{n-1}}$ ще изваждаме по принципа на включването и изключването: ще намерим колко са функциите, в които две променливи са гарантирано фиктивни, в колко три променливи са гарантирано фиктивни, и така нататък, в колко функции всички n са фиктивни, и ще построим израз с алтерниращи положителни и отрицателни знаци съгласно принципа на включването и изключването. И така:

$$|\mathbf{H}^n| = \sum_{1 \le i_1 \le n} |\mathbf{H}^n(i_1)| - \sum_{1 \le i_1 < i_2 \le n} |\mathbf{H}^n(i_1, i_2)| + \sum_{1 \le i_1 < i_2 < i_3 \le n} |\mathbf{H}^n(i_1, i_2, i_3)| - \cdots + (-1)^{n-1} |\mathbf{H}^n(i_1, \ldots, i_n)|$$

Както вече видяхме, $\sum_{1 \le i_1 \le n} |\mathbf{H}^n(i_1)| = \binom{n}{1} 2^{2^{n-1}}$. Аналогично, $\sum_{1 \le i_1 < i_2 \le n} |\mathbf{H}^n(i_1, i_2)| = \binom{n}{2} 2^{2^{n-2}}$, $\sum_{1 \le i_1 < i_2 < i_3 \le n} |\mathbf{H}^n(i_1, i_2, i_3)| = \binom{n}{3} 2^{2^{n-3}}$, и така нататък. Последното събираемо по абсолютна стойност е $|\mathbf{H}^n(i_1, \ldots, i_n)| = |\mathbf{H}^n(1, \ldots, n)| = \binom{n}{n} 2^{2^{n-n}} = 1 \times 2^{2^0} = 2$.

И така, броят на функциите с поне една фиктивна променлива е:

$$|\mathbf{H}^n| = \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} 2^{2^{n-k}}$$

Тогава броят на функциите без фиктивни променливи е:

$$|\mathbf{G}^n| = 2^{2^n} - \left(\sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} 2^{2^{n-k}} \right) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} 2^{2^{n-k}}$$

□

Задача 12. Дадени са булевите функции $f = 1011$ и $g = 1001$. Да се намери каноничното представяне на функцията $h(x_2, x_4, x_3) = f(g(x_4, x_3), x_2)$.

Решение: Имената на променливите са дадени по този начин за объркване. С просто преименуване получаваме еквивалентен израз $h(x_1, x_2, x_3) = f(g(x_2, x_3), x_1)$. Каквито и имена на променливи да ползваме, става дума за 3 променливи и таблицата на търсената функция трябва да има 8 реда:

x_1	x_2	x_3	$h(x_1, x_2, x_3)$
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

Това, че f и g са дадени без имена на променливите няма никакво значение. Очевидно $f(00) = 1$, $g(00) = 1$, $f(01) = 0$ и така нататък. Функцията h е дефинирана като $h(x_1, x_2, x_3) = f(g(x_2, x_3), x_1)$. Заместваме в таблицата:

x_1	x_2	x_3	$f(g(x_2, x_3), x_1)$
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

Ще пресмятаме отвътре навън, тъй като тази функция е композиция на $g(x_2, x_3)$ на мястото на първата променлива на f. И така, да видим какви са стойностите на $g(x_2, x_3)$ в таблицата. Те са еднозначно определени, защото на всеки ред x_2 и x_3 си имат някакви стойности, а от каноничната дефиниция на g знаем какви са функционалните ѝ стойности върху всеки вход.

x_1	x_2	x_3	$g(x_2, x_3)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Тук никъде не ползвахме най-лявата колона, защото g не зависеше от x_1. В следващата стъпка от решението пък няма да ползваме колоните на x_2 и x_3, защото f зависи непосредствено само от стойностите на g и от x_1:

x_1	x_2	x_3	$g(x_2, x_3)$	$f(g(x_2, x_3), x_1)$
0	0	0	1	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

За да се убедим, че е така, да си припомним дефиницията $f = 1011$. Функцията f е нула тогава и само тогава, когато входът е 01. На горните четири реда $x_1 = 0$, а в израза $f(g(x_2, x_3), x_1)$, x_1 е втората променлива, така че на горните четири реда функцията е четири единици. На долните четири реда, $x_1 = 1$, така че функцията просто повтаря стойностите на $g(x_2, x_3)$. □

Задача 13. Нека $f = 1000$ и $g = 0111$. Да се намери каноничната форма на функцията $h(x_1, x_2, x_3, x_4) = f(x_1, x_2) \wedge g(x_3, x_4)$.

Решение:

x_1	x_2	x_3	x_4	$f(x_1, x_2)$	$g(x_3, x_4)$	$f(x_1, x_2) \wedge g(x_3, x_4)$
0	0	0	0	1	0	0
0	0	0	1	1	1	1
0	0	1	0	1	1	1
0	0	1	1	1	1	1
0	1	0	0	0	0	0
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	0	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	1	0
1	1	0	0	0	0	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	0	1	0

И така, $h = 0111\,0000\,0000\,0000$. □

Задача 14. Дадени са булеви функции $f(x, y, z) = 11100111$ и $g(x, y, w, z) = 0011000110100101$.

- Напишете булевата функция $g(f(x, y, x), f(x, y, y), x, y)$ в каноничен вид.
- Намерете нейната СъвДНФ.

Решение: Забележете, че g е функция на 4 аргумента, но $g(f(x, y, x), f(x, y, y), x, y)$ е функция на 2 променливи, наречени x и y. Да я наречем $q(x, y)$. Търси се каноничния запис на q като булев вектор от четири булеви стойности.

$f(x, y, x)$ е функция на две променливи. Да я наречем $u(x, y)$. $u(00) = f(000) = 1$. $u(01) = f(010) = 1$. $u(10) = f(101) = 1$. $u(11) = f(111) = 1$.

$f(x, y, y)$ също е функция на две променливи. Да я наречем $v(x, y)$. $v(00) = f(000) = 1$. $v(01) = f(011) = 0$. $v(10) = f(100) = 0$ $v(11) = f(111) = 1$.

Определихме напълно u и v. Търсената $g(f(x,y,x), f(x,y,y), x, y)$ е $q(u(x,y), v(x,y), x, y)$. $q(00) = g(u(00), v(00), 0, 0) = g(1100) = 0$. $q(01) = g(u(01), v(01), 0, 1) = g(1001) = 0$. $q(10) = g(u(10), v(10), 1, 0) = g(1010) = 1$. $q(11) = g(u(11), v(11), 1, 1) = g(1111) = 1$. Тогава q е булевата функция 0011. Нейната СъвДНФ е $x\,\overline{y} \vee x\,y$. □

Задача 15. Дадени са следните функции чрез формули:

$$f(x,y,z) = xy \oplus xz \oplus yz$$
$$g(x,y,z) = (x \rightarrow y) \oplus ((y \rightarrow z) \oplus (z \rightarrow y))$$
$$h(x,y,z) = (x \rightarrow y)(y \rightarrow z) \rightarrow (x \rightarrow z)$$

Да се намерят каноничните представяния на функциите.

Отговор:

$$f(x,y,z) = 00010111$$
$$g(x,y,z) = 10010101$$
$$h(x,y,z) = 11111111$$

□

Задача 16. Еквивалентни ли са следните две формули:

$$((x \oplus y) \rightarrow (x \vee y))((\overline{x} \rightarrow y) \rightarrow (x \oplus y))$$
$$x|y$$

Решение: Да, еквивалентни са. Ще решим задачата с табличния метод (таблиците не са показани). Първата формула е конюнкция от $\tilde{1} = ((x \oplus y) \rightarrow (x \vee y))$ и $1110 = ((\overline{x} \rightarrow y) \rightarrow (x \oplus y))$, която е очевидно 1110. А долната формула—чертата на Sheffer—е на функцията 1110 по дефиниция. □

Задача 17. Докажете чрез еквивалентни преобразувания следните еквивалентности:

- $x \rightarrow (xy \rightarrow ((x \rightarrow y) \rightarrow y)z) = y \rightarrow (x \rightarrow z)$.
- $x \rightarrow (y \rightarrow z) = (x \rightarrow y) \rightarrow (x \rightarrow z)$.
- $\overline{x}(y \oplus z) = \overline{x \vee ((y \rightarrow z)(z \rightarrow y))}$.

Разрешените еквивалентни преобразувания са всички свойства на булевите функции на две променливи, дадени в учебника, и освен това свойството на импликацията $x \rightarrow y = \overline{x} \vee y$ и свойството на сумата по модул две $x \oplus y = \overline{x}y \vee x\overline{y}$, свойството на еквивалентността $x \equiv y = xy \vee \overline{x}\overline{y}$, свойството на чертата на Sheffer $x|y = \overline{xy}$ и свойството на стрелката на Peirce $x \downarrow y = \overline{x \vee y}$.

Решение: Ще покажем само последната еквивалентност, която е

$$\overline{x}(y \oplus z) = \overline{x \vee \underbrace{((y \rightarrow z)(z \rightarrow y))}_{A}}$$

Да разгледаме израза $A = (y \to z)(z \to y)$. В сила е:

$$
\begin{aligned}
A &= (y \to z)(z \to y) \quad \text{// свойство на импликацията}\\
&= (\overline{y} \vee z)(\overline{z} \vee y) \quad \text{// дистрибутивност на конюнкцията над дизюнкцията}\\
&= \overline{y}\,\overline{z} \vee \underbrace{\overline{y}y}_{0} \vee \underbrace{z\overline{z}}_{0} \vee zy\\
&= yz \vee \overline{y}\,\overline{z}
\end{aligned}
$$

И така, дясната страна е еквивалентна на

$$
\begin{aligned}
&\overline{x \vee yz \vee \overline{y}\,\overline{z}} = \quad \text{// De Morgan}\\
&\overline{x}\,\overline{yz}\,\overline{\overline{y}\,\overline{z}} = \quad \text{// De Morgan}\\
&\overline{x}(\overline{y} \vee \overline{z})(\overline{\overline{y}} \vee \overline{\overline{z}}) = \overline{x}(\overline{y} \vee \overline{z})(y \vee z) = \overline{x}(\overline{y}y \vee \overline{y}z \vee \overline{z}y \vee \overline{z}z) =\\
&\overline{x}(\overline{y}z \vee \overline{z}y) = \overline{x}(y \oplus z)
\end{aligned}
$$

□

Задача 18. Докажете чрез еквивалентни преобразувания, че x_1 е фиктивна променлива в следните функции:

- $f(x_1, x_2) = (x_2 \to x_1)(x_2 \downarrow x_2)$.
- $g(x_1, x_2) = (x_2 \equiv x_1) \vee (x_1 | x_2)$.
- $h(x_1, x_2, x_3) = ((x_1 \oplus x_2) \to x_3)(\overline{x_3 \to x_2})$.

Решение:

$$
\begin{aligned}
f(x_1, x_2) &= (x_2 \to x_1)(x_2 \downarrow x_2) = (x_2 \to x_1)\left(\overline{x_2 \vee x_2}\right) = (x_2 \to x_1)\overline{x_2} =\\
&(\overline{x_2} \vee x_1)\overline{x_2} = \overline{x_2}\,\overline{x_2} \vee x_1\overline{x_2} = \overline{x_2} \vee x_1\overline{x_2} = \overline{x_2}(1 \vee x_1) = \overline{x_2}
\end{aligned}
$$

$$
\begin{aligned}
g(x_1, x_2) &= (x_2 \equiv x_1) \vee (x_1 | x_2) = (x_1x_2 \vee \overline{x_1x_2}) \vee (\overline{x_1x_2}) = x_1x_2 \vee \overline{x_1x_2} \vee \overline{x_1} \vee \overline{x_2} =\\
&x_1x_2 \vee \overline{x_1}(\overline{x_2} \vee 1) \vee \overline{x_2} = x_1x_2 \vee \overline{x_1}1 \vee \overline{x_2} = x_1x_2 \vee \overline{x_1} \vee \overline{x_2} =\\
&x_1x_2 \vee \overline{x_1x_2} = \quad \text{// нека } x_1x_2 = p\\
&p \vee \overline{p} = 1
\end{aligned}
$$

$$
\begin{aligned}
h(x_1, x_2, x_3) &= ((x_1 \oplus x_2) \to x_3)(x_3 \to x_2) = (\overline{x_1 \oplus x_2} \vee x_3)(\overline{\overline{x_3} \vee x_2}) =\\
&(\overline{x_1 \oplus x_2} \vee x_3)(\overline{\overline{x_3}}\,\overline{x_2}) = (\overline{x_1 \oplus x_2} \vee x_3)\overline{x_2}x_3 = (\overline{x_1\overline{x_2} \vee \overline{x_1}x_2} \vee x_3)\overline{x_2}x_3 =\\
&((\overline{x_1\overline{x_2}}\,\overline{\overline{x_1}x_2}) \vee x_3)\overline{x_2}x_3 = ((\overline{x_1} \vee \overline{\overline{x_2}})(\overline{\overline{x_1}} \vee \overline{x_2}) \vee x_3)\overline{x_2}x_3 =\\
&((\overline{x_1} \vee x_2)(x_1 \vee \overline{x_2}) \vee x_3)\overline{x_2}x_3 = ((\overline{x_1}x_1 \vee \overline{x_1}\,\overline{x_2} \vee x_2x_1 \vee x_2\overline{x_2}) \vee x_3)\overline{x_2}x_3 =\\
&((\overline{x_1}\,\overline{x_2} \vee x_2x_1) \vee x_3)\overline{x_2}x_3 = (\overline{x_1}\,\overline{x_2} \vee x_2x_1 \vee x_3)\overline{x_2}x_3 =\\
&\overline{x_1}\,\overline{x_2}\,\overline{x_2}x_3 \vee x_2x_1\overline{x_2}x_3 \vee x_3\overline{x_2}x_3 = \overline{x_1}\,\overline{x_2}x_3 \vee 0 \vee \overline{x_2}x_3 = \overline{x_1}\,\overline{x_2}x_3 \vee \overline{x_2}x_3 =\\
&(\overline{x_1} \vee 1)\overline{x_2}x_3 = 1\overline{x_2}x_3 = \overline{x_2}x_3
\end{aligned}
$$

□

Задача 19. Докажете, че всяка симетрична булева функция, различна от константа, има само съществени променливи.

Решение: Да наречем тази функция f. Нека n е броят на нейните променливи. Щом f не е константа, f има стойност 0 върху поне един вектор и стойност 1 върху поне един вектор. Съгласно разсъжденията в решението на Задача 5, за всеки слой на n-мерния хиперкуб, f има една и съща стойност върху всички вектори от този слой. Лесно се вижда, че в хиперкуба има съседни слоеве (тоест, единият има една единица повече от другия), такива че f има стойност 0 върху единия от тях и стойност 1 върху другия от тях.

Без ограничение на общността, нека f има стойност 0 върху всички вектори от слой k и стойност 1 върху всички вектори от слой $k+1$, за някое k, такова че $0 \leq k \leq n-1$. Ще докажем едно помощно твърдение, чиято важност налага да го обособим като лема. Използваме контрапозитивното твърдение† на Лема 1 и търсеният резултат следва веднага.

Лема 1. Ако $f \in \mathcal{F}^n$ има поне една фиктивна променлива, то за всеки два съседни слоя L_k и L_{k+1} на n-мерния хиперкуб съществува вектор $\mathbf{a} \in L_k$ и съществува вектор $\mathbf{b} \in L_{k+1}$, такива че $f(\mathbf{a}) = f(\mathbf{b})$.

Доказателство: Нека x_i е фиктивна променлива. По дефиниция:

$$f(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, x_{i+2}, \ldots, x_n) = f(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, x_{i+2}, \ldots, x_n)$$

за всички булеви $(n-1)$-вектори $x_1, x_2, \ldots, x_{i-1}, x_{i+1}, x_{i+2}, \ldots, x_n$.

Да разгледаме произволно k, такова че $0 \leq k \leq n-1$. Да разгледаме произволен булев $(n-1)$-вектор

$$x_1, x_2, \ldots, x_{i-1}, x_{i+1}, x_{i+2}, \ldots, x_n\,,$$

който има точно k единици. Тогава n-векторът

$$\mathbf{a} = x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, x_{i+2}, \ldots, x_n$$

има точно k единици, а n-векторът

$$\mathbf{b} = x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, x_{i+2}, \ldots, x_n$$

има точно $k+1$ единици. Тогава $\mathbf{a} \in L_k$, $\mathbf{b} \in L_{k+1}$ и $f(\mathbf{a}) = f(\mathbf{b})$. **QED**

□

Следващата задача ползва релацията $\preccurlyeq$, дефинирана в (2).

Задача 20. Да разгледаме някаква $f \in \mathcal{F}^n$, такава че съществуват k вектори $\mathbf{a}_1$, $\mathbf{a}_2$, …, $\mathbf{a}_k$ за някакво $k \geq 2$, такива че

$$\mathbf{a}_1 \preccurlyeq \mathbf{a}_2 \preccurlyeq \cdots \preccurlyeq \mathbf{a}_k$$
$$f(\mathbf{a}_1) \neq f(\mathbf{a}_2) \neq \cdots \neq f(\mathbf{a}_k)$$

Да се докаже, че функцията има поне $k-1$ съществени променливи.

† Контрапозитивното е "Ако съществуват съседни слоеве L_k и L_{k+1} на хиперкуба, такива че $\forall \mathbf{a} \in L_k \forall \mathbf{b} \in L_{k+1} : f(\mathbf{a}) \neq f(\mathbf{b})$, то f няма фиктивни променливи".

Решение: Релацията $\preccurlyeq$ е рефлексивна, но от второто ограничение следва, че векторите са два по два различни. Да си припомним и релацията $\prec$, дефинирана в (3). Очевидно, по отношение на $\preccurlyeq$ има верига

$$\mathbf{a}_1 \prec \mathbf{b}_{1,1} \prec \cdots \prec \mathbf{b}_{1,t_1} \prec \mathbf{a}_2 \prec \mathbf{b}_{2,1} \prec \cdots \prec \mathbf{b}_{2,t_2} \prec \cdots \prec \mathbf{a}_{k-1} \prec \mathbf{b}_{k-1,1} \prec \cdots \prec \mathbf{b}_{k-1,t_{k-1}} \prec \mathbf{a}_k$$

Твърдим, че в тази верига има поне $k-1$ различни съседни двойки вектори[†], такива че функцията има противоположни стойности върху векторите от всяка двойка. Това твърдение е очевидно и няма да го доказваме. Векторите от всяка двойка се различават в точно една позиция, следователно са вектори от два съседни слоя на хиперкуба. Прилагаме контрапозитивното твърдение на Лема 1 и виждаме, че функцията има $k-1$ фиктивни променливи.

Факта, че всяка двойка двойки задава **различни** фиктивни променливи, поради което заключаваме, че променливите не може да са по-малко от $k-1$, е очевиден. □

Задача 21. Нека $f, g \in \mathbb{F}_2^n$ са такива, че $f(\mathbf{a}) \oplus g(\mathbf{a}) = 1$ за точно нечетен брой вектори $\mathbf{a} \in \{0,1\}^n$. Да се докаже, че всяка променлива е съществена за поне едната от функциите f и g.

Решение: Ограничението "$f(\mathbf{a}) \oplus g(\mathbf{a}) = 1$ за точно нечетен брой вектори" е същото като "f и g се различават върху точно нечетен брой вектори" – това следва тривиално от дефиницията на функцията $\oplus$.

И така, двете функции имат противоположни стойности върху подмножество на $\{0,1\}^n$ с нечетна мощност. Да допуснем, че съществува променлива x_i, която е фиктивна и за двете функции. По дефиниция:

$$\begin{aligned} f(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, x_{i+2}, \ldots, x_n) &= f(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, x_{i+2}, \ldots, x_n) \\ g(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, x_{i+2}, \ldots, x_n) &= g(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, x_{i+2}, \ldots, x_n) \end{aligned}$$

за всяко раздаване на 0 и 1 на променливите x_1, x_2, …, x_{i-1}, x_{i+1}, x_{i+2}, …, x_n.

Да си представим и двете функции едновременно върху хиперкуба – да си представим n-мерния хиперкуб и до всеки негов връх, стойността на функцията f в червено и стойността на функцията g в синьо.

Да срежем хиперкуба в i-тата размерност. Получаваме два хиперкуба, всеки от размерност $n-1$. За произволен връх u от единия получен $(n-1)$-мерен хиперкуб, ако v е неговият съответен връх[‡] в другия $(n-1)$-мерен хиперкуб, ясно е, че $f(u) = f(v)$ и $g(u) = g(v)$ – това е заради фиктивността на x_i по отношение и на f, и на g.

Следователно, броят на върховете, върху които f и g имат **една и съща стойност** в единия $(n-1)$-мерен хиперкуб е равен на броя на върховете, върху които f и g имат **една и съща стойност** в другия $(n-1)$-мерен хиперкуб. А оттук следва, че броят на върховете, върху които f и g имат **различна стойност** в единия $(n-1)$-мерен хиперкуб е равен на броя на върховете, върху които f и g имат **различна стойност** в другия $(n-1)$-мерен хиперкуб. Но броят на върховете, върху които f и g се различават в оригиналния (преди срязването) n-мерен хиперкуб, е равен на сумата от броя на върховете, върху f и g се различават върху единия получен $(n-1)$-мерен хиперкуб и броя на върховете, върху f и g се различават върху другия получен $(n-1)$-мерен хиперкуб. Щом двете събираеми са равни, тяхната сума е четно число. Тогава броят на върховете, върху които f и g се различават в оригиналния (преди срязването) n-мерен хиперкуб, е четна. Което противоречи на условието на задачата. □

[†] Две такива двойки може да имат общ елемент, въпреки че са различни като двойки.

[‡] "Съответен връх" означава, че има същия етикет, тоест $(n-1)$-мерен булев вектор.

Задача 22. Докажете чрез еквивалентни преобразувания, че следните две формули не са еквивалентни:

$$
\begin{aligned}
U &= (x \downarrow \overline{y}) \rightarrow \big(\overline{x}\,\overline{z} \rightarrow ((\overline{x}|(y \equiv z)) \vee (\overline{xy} \vee z))\big)\\
V &= ((x \rightarrow y)|(x \downarrow (y\overline{z})) \vee \overline{yz}
\end{aligned}
$$

Разрешените еквивалентни преобразувания са тези от Задача 17.

Решение: От една страна:

$$
\begin{aligned}
U &= \overline{x \downarrow \overline{y}} \vee \big(\overline{\overline{x}\,\overline{z}} \vee ((\overline{x}|(y \equiv z)) \vee (\overline{xy} \vee z))\big)\\
&= \overline{\overline{x \vee \overline{y}}} \vee \overline{\overline{x}\,\overline{z}} \vee ((\overline{x}|(y \equiv z)) \vee (\overline{xy} \vee z))\\
&= x \vee \overline{y} \vee \overline{\overline{x}\,\overline{z}} \vee ((\overline{x}|(y \equiv z)) \vee \overline{x} \vee \overline{y} \vee z = \quad \text{// понеже е от вида } x \vee \cdots \overline{x} \vee \cdots\\
&= 1
\end{aligned}
$$

От друга страна:

$$
\begin{aligned}
V &= \overline{(\overline{x} \vee y)(\overline{x \vee y\overline{z}})} \vee \overline{yz}\\
&= (\overline{\overline{x} \vee y}) \vee (\overline{\overline{x \vee y\overline{z}}}) \vee \overline{y} \vee \overline{z}\\
&= \overline{\overline{x}}\,\overline{y} \vee x \vee y\overline{z} \vee \overline{y} \vee \overline{z}\\
&= x\overline{y} \vee 1\overline{x} \vee y\overline{z} \vee 1\overline{z} \vee \overline{y}\\
&= x(\overline{y} \vee 1) \vee \overline{z}(y \vee 1) \vee \overline{y} = x \vee \overline{y} \vee \overline{z}
\end{aligned}
$$

Показахме, че V е еквивалентна на $x \vee \overline{y} \vee \overline{z}$. От тази проста формула лесно се вижда, че семантиката на V не е константа-единица. Следователно, U и V не са еквивалентни. □

Задача 23. Намерете СъвДНФ на булевата функция $f = 01110100$.

Решение: Имената на променливите не са уточнени, така че имаме свобода да си ги изберем. Избираме x, y и z, като се ползва традиционната наредба x-преди-y и y-преди-z. Тогава таблицата на функцията е:

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

СъвДНФ е:

$$\overline{x}\,\overline{y}\,z \vee \overline{x}\,y\,\overline{z} \vee \overline{x}\,y\,z \vee x\,\overline{y}\,z$$

□

Задача 24. Намерете СъвДНФ на следните булеви функции:

A) $(x_1 \oplus x_2) \rightarrow x_2x_3$

Б) $f = 01101100$

В) $g = 0001\,1101\,1001\,1011$

Г) $\overline{x_1\overline{x_2} \rightarrow \overline{x_3}}$

Д) $(x|y)\overline{z}$

Е) $xy \equiv (y \equiv z)$

Ако променливите не са именувани явно, допуснете подходящо именуване с, например, x, y, z или x_1, x_2, x_3 и така нататък.

Отговори и решения:

А) $(x_1 \oplus x_2) \rightarrow x_2x_3 = \overline{x_1}\,\overline{x_2}\,\overline{x_3} \vee \overline{x_1}\,\overline{x_2}\,x_3 \vee \overline{x_1}\,x_2\,x_3 \vee x_1\,x_2\,\overline{x_3} \vee x_1\,x_2\,x_3$

Б) $f(x, y, z) = \overline{x}\,\overline{y}\,z \vee \overline{x}\,y\,\overline{z} \vee x\,\overline{y}\,\overline{z} \vee x\,\overline{y}\,z$

В) $g(x_1, x_2, x_3, x_4) = \overline{x_1}\,\overline{x_2}\,x_3\,x_4 \vee \overline{x_1}\,x_2\,\overline{x_3}\,\overline{x_4} \vee \overline{x_1}\,x_2\,\overline{x_3}\,x_4 \vee \overline{x_1}\,x_2\,x_3\,x_4 \vee x_1\,\overline{x_2}\,\overline{x_3}\,\overline{x_4} \vee x_1\,\overline{x_2}\,x_3\,x_4 \vee x_1\,x_2\,\overline{x_3}\,\overline{x_4} \vee x_1\,x_2\,x_3\,\overline{x_4} \vee x_1\,x_2\,x_3\,x_4$

Г) $\overline{x_1\overline{x_2} \rightarrow \overline{x_3}} = \overline{\overline{x_1\overline{x_2}} \vee \overline{x_3}} = \overline{\overline{x_1} \vee x_2 \vee \overline{x_3}} = \overline{\overline{x_1}}\,\overline{x_2}\,\overline{\overline{x_3}} = x_1\overline{x_2}x_3$. В този случай намерхиме СъвДНФ с еквивалентни преобразувания. По-точно казано, извършихме редица от еквивалентни преобразувания, резултатът от които е израз, който е СъвДНФ по **формални** причини: има точно такава форма, каквато трябва да има една СъвДНФ.

Д) $(x|y)\overline{z} = \overline{xy}\,\overline{z}$. За съжаление, полученият израз **не е** СъвДНФ, защото няма желаната форма! За да е СъвДНФ с една пълна конюнктивна клауза, изразът трябва да е конкатенация от литерали, а този израз не е такъв, защото $\overline{xy}$ нито е литерал, нито е конкатенация от литерали. Ако искаме да решим подзадачата с еквивалентни преобразувания, трябва да продължим. $\overline{xy}\,\overline{z} = (\overline{x} \vee \overline{y})\overline{z} = \overline{x}\,\overline{z} \vee \overline{y}\,\overline{z} = \overline{x}\,\overline{z}(y \vee \overline{y}) \vee \overline{y}\,\overline{z}(x \vee \overline{x}) = \overline{x}y\overline{z} \vee \overline{x}\,\overline{y}\,\overline{z} \vee x\,\overline{y}\,\overline{z} \vee \overline{x}\,\overline{y}\,\overline{z} = \overline{x}y\overline{z} \vee \overline{x}\,\overline{y}\,\overline{z} \vee x\,\overline{y}\,\overline{z}$

Е) $\overline{x}y\overline{z} \vee x\overline{y}z \vee xyz \vee \overline{x}\,\overline{y}z$

□

Задача 25. Колко СъвДНФ има над n променливи?

Решение: Броят на пълните конюнктивни клаузи е точно 2^n, защото във всяка от тях участват точно n литерала, а за всеки от тях има точно 2 възможности независимо от останалите. Всяка СъвДНФ се определя еднозначно от това, кои пълни конюнктивни клаузи участват. Но отговорът 2^{2^n} за броя на всички СъвДНФ не е точен, понеже не може да няма нито една пълна конюнктивна клауза. С други думи, празната формула (празният стринг) не е СъвДНФ по дефиниция. Отговорът е $2^{2^n} - 1$.

Можем да го получим и с други разсъждения: всяка булева функция на n променливи без константа-нула има точно една СъвДНФ, а на различни СъвДНФ съответстват различни булеви функции. □

Задача 26. Колко ДНФ има над n променливи?

Решение: Трябва да съобразим колко конюнктивни клаузи има. В конюнктивните клаузи има три възможности за всяка променлива (а не две, както беше при пълните конюнктивни клаузи) – променливата може да участва като положителен литерал, като отрицателен литерал или изобщо да не участва. Това означава 3^n възможности, но ако броим и възможността да няма участващи променливи изобщо, тоест празната последователност от литерали (празният стринг). Но ние искаме всяка пълна конюнктивна клауза да е непразна, тоест да има поне един литерал, така че възможността да няма нито един литерал отпада и възможностите са $3^n - 1$.

Всяка от пълните конюнктивни клаузи може да участва или да не участва, но трябва да има поне една такава, така че общо възможностите са $2^{3^n - 1} - 1$. □

Задача 27. Намерете броя на булевите функции от $\mathcal{F}^n$, чиито СъвДНФ изпълняват условието:

1. Няма пълна конюнктивна клауза, в която броят на положителните литерали е равен на броя на отрицателните литерали.
2. Всяка пълна конюнктивна клауза има четен брой отрицателни литерали.
3. Всяка пълна конюнктивна клауза има поне два отрицателни литерала.

Решение:

1. Ако n е нечетно, то всички СъвДНФ са такива и отговорът е $2^{2^n} - 1$. Ако n е четно, отговорът е $2^p - 1$, където p е броят на n-векторите с неравен брой нули и единици. Очевидно, $p = 2^n - q$, където q е броят на n-векторите с равен брой нули и единици. Но ние знаем, че има точно $\binom{n}{n/2}$ начина да разположим $n/2$ нули и $n/2$ единици в линейна наредба, следователно $q = \binom{n}{n/2}$, следователно $p = 2^n - \binom{n}{n/2}$, следователно отговорът е $2^{2^n - \binom{n}{n/2}} - 1$.
2. За всеки входен вектор съответната пълна конюнктивна клауза участва в СъвДНФ тстк функцията има стойност 1 върху този входен вектор. Иска се функцията да е задължително 0 върху всички входни вектори с нечетен брой нули, тоест стойността на функцията може да варира само върху векторите с четен брой нули. Отговорът е $2^{2^{n-1}} - 1$, защото точно половината† вектори, тоест $\frac{1}{2}2^n = 2^{n-1}$, са тези с четен брой нули, а от $2^{2^{n-1}}$ вадим единица заради това, че функцията константа-нула няма СъвДНФ.
3. Върху векторите с нула нули и точно една нула, функцията трябва да е задължително 0. Освен това, не може да е константа-нула. Други ограничения няма. Има един вектор с нула нули и n вектора с точно една нула. Отговорът е $2^{2^n - n - 1} - 1$.

□

Задача 28. Нека $n \geq 2$. Да се определи дължината на СъвДНФ (като брой на пълните конюнктивни клаузи) на следните булеви функции, представени чрез формули:

1. $x_1 \oplus x_2 \oplus \cdots \oplus x_n$
2. $(x_1 \vee x_2 \vee \cdots \vee x_n)(\overline{x_1} \vee \overline{x_2} \vee \cdots \vee \overline{x_n})$

†Това се извежда тривиално, имайки предвид резултата от комбинаториката, че броят на подмножествата с четна мощност е равен на броя на подмножествата с нечетна мощност.

3. $\bigvee_{1 \leq i < j \leq n} x_i x_j$

4. $\bigwedge_{1 \leq i < j \leq n} (x_i \rightarrow x_j)$

Решение: Задачата е същата като "колко единици съдържа каноничното представяне на функцията?". Каноничното представяне на всяка от тези четири функции има лесна за определяне форма. Трябва да разберем каква е закономерността на каноничното представяне на всяка от тях. Сторим ли това, решението става тривиално.

1. Функцията е сума по модул две на всички променливи. Очевидно, тази сума е 1 точно върху тези вектори, които имат нечетен брой единици. Както вече казахме в решението на Задача 27, векторите с нечетен брой единици са точно половината, тоест $\frac{1}{2}2^n = 2^{n-1}$. Отговорът е точно 2^{n-1}. Забележете, че **не е** $2^{2^{n-1}}$, защото върху въпросните вектори стойностите на функцията не варират, а са само единици.

2. Лесно се вижда, че функцията е 0 върху точно два вектора: $00 \cdots 0$ и $11 \cdots 1$. Отговорът е $2^n - 2$.

3. Функцията има стойност 1 точно върху тези вектори, които имат поне две единици в себе си. Векторите, които нямат поне две единици в себе си, са тези с нула единици (само един) и с точно една единица (n такива). Отговорът е $2^n - n - 1$.

4. Може би е по-лесно да се съобрази каква е формата на каноничното представяне, ако използваме различно представяне на същата функция:

$$\bigwedge_{1 \leq i < j \leq n} (x_i \rightarrow x_j) = \bigwedge_{1 \leq i < j \leq n} (\overline{x_i} \vee x_j)$$

Която и от двете формули да разгледаме, виждаме, че функцията е нула върху точно тези вектори, които имат поне една 1, която е вляво от поне една 0[†]. Следователно, функцията е единица точно върху векторите $00 \cdots 00$, $00 \cdots 01$, $00 \cdots 11$, $\ldots$, $01 \cdots 11$, $11 \cdots 11$. Те са точно $n + 1$, и това е отговорът. □

Задача 29. Нека $f(x_1, x_2, \ldots, x_n)$ и $g(y_1, y_2, \ldots, y_n)$ са булеви функции, такива че СъвДНФ на f има k_1 пълни конюнктивни клаузи, а на g, k_2 пълни конюнктивни клаузи. Да се определи дължината, в брой пълни конюнктивни клаузи, на:

$$\begin{aligned} h(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n) &= fg \\ t(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n) &= f \vee g \end{aligned}$$

Решение: Да разгледаме първо fg. Забележете, че това е функция **не** на n, а на $2n$ променливи. Нейната таблица е показана схематично на следната таблица. Да си представим, че върху входните вектори, всеки с дължина $2n$, които са общо 2^{2n} на брой, е дефинирано групиране в нещо като правоъгълници с размери $n \times 2^n$, които на таблицата са показани в лявата страна, оцветени в розово и зелено. Всеки зелен правоъгълник огражда една група от стойности на y-променливите, като групите са 2^n общо и съдържанието им е едно и също: всяка група съдържа точно всички n-вектори в лексикографски ред. Розовите правоъгълници ограждат групи от стойности на x-променливите, като тези групи пак са 2^n общо и всяка има 2^n n-вектора, но сега всяка група съдържа 2^n копия на един и същи n-вектор.

Тъй като функцията g зависи от y-стойностите, а функцията f, от x-стойностите, в дясната страна, където са функционалните стойности, наблюдаваме следните закономерности.

† Тоест, в които има $x_i = 1$ и $x_j = 0$ при $i < j$; тоест, които са от вида $\cdots 1 \cdots 0 \cdots$.

Колоната на g се състои от 2^n подколони, всяка от които е копие на един и същи pattern, който има точно k_2 единици (по условие g има СъвДНФ с точно k_2 пълни конюнктивни клаузи). Точно какво е съдържанието на тази повтаряща се колона зависи от конкретиката на g. В таблицата "$\tilde{g}_i$" е кратък запис за $g(\tilde{y}_i)$, където $\tilde{y}_i$ е i-ият вектор от y-векторите, за $0 \leq i \leq 2^n - 1$.

От друга страна, колоната на f се състои от 2^n подколони, всяка от които съдържа 2^n копия на една и съща стойност, а именно функционалната стойност на f върху i-ия от x-векторите за $0 \leq i \leq 2^n - 1$, която е отбелязана с $\tilde{f}_i$. Очевидно точно k_1 от тези подколони са от единици. Кои точно, зависи от конкретиката на f.

И така, функцията h е конюнкция от f и g. Колоната на h не е изобразена, но лесно може да си я представим написана най-вдясно, с височина 2^{2n}, състояща от поелементни конюнкции от колоните на g и f. Тривиално е да се съобрази, че ако разбием колоната на h на 2^n подколони аналогично на разбиванията на колоните на g и f, точно $2^n - k_1$ от подколоните ще са само от нули, защото колоната на f има точно $2^n - k_1$ от подколони от нули, които в поелементните конюнкции "нулират" стойността на h върху дадения $2n$-вектор независимо от стойностите на g върху него. А останалите k_1 на брой подколони на f са само от единици. За всяка от тях, функцията h получава точно k_2 единици заради поелементната конюнкция.

Общо, h има точно $k_1 \times k_2$ единици.

Сега да разгледаме функцията t. Съображенията са аналогични и таблицата е същата, но сега търсим мощността не на сечение, а на обединение. Колоната на g има общо $k_2 \times 2^n$ единици, а колоната на f има общо $k_1 \times 2^n$ единици. От $k_1 2^n + k_2 2^n = (k_1 + k_2)2^n$ трябва да извадим, съгласно принципа на включването и изключването, мощността на сечението, която, както вече установихме, е $k_1 k_2$. Отговорът е $(k_1 + k_2)2^n - k_1 k_2$.

$x_1, x_2, \ldots, x_n$		$y_1, y_2, \ldots, y_n$		g		f		
$0, 0, \ldots, 0, 0$ $0, 0, \ldots, 0, 0$ $0, 0, \ldots, 0, 0$ $\ldots\ldots\ldots\ldots$ $0, 0, \ldots, 0, 0$ $0, 0, \ldots, 0, 0$	2^n копия на $00\cdots00$	$0, 0, \ldots, 0, 0$ $0, 0, \ldots, 0, 1$ $0, 0, \ldots, 1, 0$ $\ldots\ldots\ldots\ldots$ $1, 1, \ldots, 1, 0$ $1, 1, \ldots, 1, 1$	всички n-вектори, лексикографски	$\tilde{g}_0$ $\tilde{g}_1$ $\tilde{g}_2$ $\ldots$ $\tilde{g}_{2^n-2}$ $\tilde{g}_{2^n-1}$	pattern с k_2 единици	$\tilde{f}_0$ $\tilde{f}_0$ $\tilde{f}_0$ $\ldots$ $\tilde{f}_0$ $\tilde{f}_0$	2^n еднакви ст-сти	от всички тези, 2^n на брой, групи, всяка от 2^n еднакви ст-сти, точно k_1 групи са от единици
$0, 0, \ldots, 0, 1$ $0, 0, \ldots, 0, 1$ $0, 0, \ldots, 0, 1$ $\ldots\ldots\ldots\ldots$ $0, 0, \ldots, 0, 1$ $0, 0, \ldots, 0, 1$	2^n копия на $00\cdots01$	$0, 0, \ldots, 0, 0$ $0, 0, \ldots, 0, 1$ $0, 0, \ldots, 1, 0$ $\ldots\ldots\ldots\ldots$ $1, 1, \ldots, 1, 0$ $1, 1, \ldots, 1, 1$	всички n-вектори, лексикографски	$\tilde{g}_0$ $\tilde{g}_1$ $\tilde{g}_2$ $\ldots$ $\tilde{g}_{2^n-2}$ $\tilde{g}_{2^n-1}$	същият pattern с k_2 ед.	$\tilde{f}_1$ $\tilde{f}_1$ $\tilde{f}_1$ $\ldots$ $\tilde{f}_1$ $\tilde{f}_1$	2^n еднакви ст-сти	
$\ldots\ldots\ldots\ldots$ $\ldots\ldots\ldots\ldots$ $\ldots\ldots\ldots\ldots$ $\ldots\ldots\ldots\ldots$		$\ldots\ldots\ldots\ldots$ $\ldots\ldots\ldots\ldots$ $\ldots\ldots\ldots\ldots$ $\ldots\ldots\ldots\ldots$		$\ldots\ldots$ $\ldots\ldots$ $\ldots\ldots$ $\ldots\ldots$		$\ldots\ldots$ $\ldots\ldots$ $\ldots\ldots$ $\ldots\ldots$		
$1, 1, \ldots, 1, 1$ $1, 1, \ldots, 1, 1$ $1, 1, \ldots, 1, 1$ $\ldots\ldots\ldots\ldots$ $1, 1, \ldots, 1, 1$ $1, 1, \ldots, 1, 1$	2^n копия на $11\cdots11$	$0, 0, \ldots, 0, 0$ $0, 0, \ldots, 0, 1$ $0, 0, \ldots, 1, 0$ $\ldots\ldots\ldots\ldots$ $1, 1, \ldots, 1, 0$ $1, 1, \ldots, 1, 1$	всички n-вектори, лексикографски	$\tilde{g}_0$ $\tilde{g}_1$ $\tilde{g}_2$ $\ldots$ $\tilde{g}_{2^n-2}$ $\tilde{g}_{2^n-1}$	същият pattern с k_2 ед.	$\tilde{f}_{2^n-1}$ $\tilde{2}_{2^n-1}$ $\tilde{f}_{2^n-1}$ $\ldots$ $\tilde{f}_{2^n-1}$ $\tilde{f}_{2^n-1}$	2^n еднакви ст-сти	

$\square$

3 Благодарности

Авторът благодари много на **Добромир Кралчев** за многобройните корекции на граматически и стилистични грешки, както и за корекциите на грешки в решенията на две от задачите. Благодарности на **Стоян Томицин** за откритата грешка в условието на Задача 13. Благодарности на **Християн Валентинов Тонев** за откритите и коригирани грешки в решенията на осем от задачите.

Авторът благодари отново на **Добромир Кралчев** за многобройните граматични и стилистични корекции при повторен прочит.