

## 4.2 Формули и суперпозиции

Всяка двоична функция  $f(x_1, x_2, \dots, x_n)$  може да се представи с таблица по следния начин

Таблица 4.4

$x_1$	$x_2$	...	$x_{n-1}$	$x_n$	$f(x_1, x_2, \dots, x_{n-1}, x_n)$
0	0	...	0	0	$f(0, 0, \dots, 0, 0)$
0	0	...	0	1	$f(0, 0, \dots, 0, 1)$
0	0	...	1	0	$f(0, 0, \dots, 1, 0)$
0	0	...	1	1	$f(0, 0, \dots, 1, 1)$
.	.	...	.	.	. . . . .
1	1	...	1	0	$f(1, 1, \dots, 1, 0)$
1	1	...	1	1	$f(1, 1, \dots, 1, 1)$

Не е трудно да се види, че табличното представяне изисква прекалено много място даже и за функции с неголям брой променливи (за функция на  $n$  променливи са необходими  $(n+1)2^n$  бита). Ако фиксираме реда на  $n$ -орките от лявата част на таблицата, то най-десният стълб ще описва напълно функцията  $f(x_1, x_2, \dots, x_n)$ . От тук следва, че за задаването на каквато и да е функция на  $n$  променливи са необходими  $2^n$  бита. На практика се налага да се изследват функции с над 100 променливи, което чрез таблици просто не е възможно. И най-големите компютри нямат достатъчно памет за вместването на подобна функция. Един друг начин (вече използван от нас) е задаването на функцията чрез формули.

**Пример 4.2.1** Функцията  $f(x_1, x_2, x_3)$  на 3 променливи, зададена чрез таблица 4.5.

Таблица 4.5

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

се описва много по-икономично с формулата  $x_1x_2x_3$ .

С помощта на апарата на формулите можем да опишем достатъчно компактно голям брой практически важни функции. Трябва да се има предвид, обаче, че за почти всички функции задаването с таблица и задаването с формула изисква примерно еднаква машинна памет.

Между понятията *формула* и *функция* има съществена разлика. Функцията е съответствие на едно множество в друго, а формулата е израз, низ от символи, които описва (ще казваме още реализира) дадена функция. Възможно е една функция да се описва от много (даже безбройно много формули). Пример за това е функцията отрицание, която се реализира от формулите  $\overline{x}, \overline{\overline{x}}, \overline{\overline{\overline{x}}}$  и т. н.

Необходимо е с подходящи дефиниции да прецизираме горните разсъждения.

Нека  $F = \{f_1(x_1, x_2, \dots, x_{n_1}), f_2(x_1, x_2, \dots, x_{n_2}), \dots\}$  е множество от функции.

**Определение 4.2.1** *Под формула над  $F$  ще разбираме:*

1. *Всички функции от  $F$ .*
2. *Всички изрази от вида*

$$f_1(\varphi_1, \varphi_2, \dots, \varphi_n),$$

където  $f_1 \in F$ , а  $\varphi_j$  е или формула над  $F$ , или буква на променлива  $x_1, x_2, \dots, x_n$ .

На пръв поглед горната дефиниция може да се стори за грешна, защото в нейната т.2 понятието формула се определя чрез формула. Разбира се, че това не е така и дефиницията е съвсем редовна. Нейният смисъл е следният

1. Дадени обекти явно обявяваме за формули - това са функциите от  $F$ .

2. Задаваме правило, чрез което от вече известните формули може да получаваме останалите формули.

Подобни дефиниции ще наричаме *рекурсивни*.

**Пример 4.2.2** Нека  $F = \{f_1(x_1), f_2(x_1, x_2)\}$ . Тогава

$$f_1(x_1), f_2(x_1, x_1), f_2(f_1(x_1), x_3), f_2(f_1(x_1), f(x_7)), \\ f_2(f_1(x_1), f_2(f_1(x_5), x_3))$$

са някои от формулите над  $F$ .

Когато  $f(x_1, x_2)$  е функция на 2 променливи, често ще пишем  $x_1 f x_2$  вместо  $f(x_1, x_2)$  - например  $x_1 + x_2$  вместо  $+(x_1, x_2)$  или  $x_1 \vee x_2$  вместо  $\vee(x_1, x_2)$  и т. н. Ако спазваме правилата за приоритетите а) - д) от предишния параграф, това ще ни позволи да записваме по-нагледно формулите, в които участват  $x_1 x_2$ ,  $x_1 \vee x_2$ ,  $x_1 + x_2$ ,  $x_1 \rightarrow x_2$  и  $\overline{x}$ .

**Пример 4.2.3** Формулата  $(x_1) \vee (x_2(x_3(\overline{x_4})))$  ще записваме съкратено с  $x_1 \vee x_2 x_3 \overline{x_4}$ .

По естествен начин на всяка формула  $\varphi$  може да съпоставим функцията  $f$ , която тя реализира. Това става със следното правило:

1. На формулите  $f_i(x_1, x_2, \dots, x_{n_i}) \in F$  съпоставяме самата функция  $f_i$ .

2. Нека на формулите  $\varphi_1, \varphi_2, \dots, \varphi_{n_i}$  сме вече съпоставили функциите  $g_1, g_2, \dots, g_{n_i}$ . Тогава на  $f_i(\varphi_1, \varphi_2, \dots, \varphi_{n_i})$  се съпоставя сложната функция  $f_1(g_1, g_2, \dots, g_{n_i})$ .

**Определение 4.2.2** Ако една функция  $f$  се реализира от поне една формула над  $F$ , то ще казваме, че  $f$  е **суперпозиция над  $F$** .

Ние често ще използваме равенства от вида  
 $f = F$ , което ще означава, че функцията  $f$  се реализира  
от формулата  $F$ ;  
 $F_1 = F_2$ , което означава, че формулите  $F_1$  и  $F_2$  реализират  
една и съща функция и такива формули  
ще наричаме *еквивалентни*.

**Определение 4.2.3** Множеството от всички суперпозиции над  $F$  ще наричаме **затворена обвивка** на  $F$  и ще го означаваме с  $[F]$ . С други думи  $[F]$  се състои от всички функции, които можем да реализираме с формули над  $F$ .

Да отбележим 4 важни очевидни свойства на затворената обвивка:

1.  $F \subset [F]$ .
2. От  $F \subset G$  следва  $[F] \subset [G]$ .
3.  $[F] \cup [G] \subset [F \cup G]$ .
4.  $[[F]] = [F]$ .

**Определение 4.2.4** Ще казваме, че множеството от двоични функции  $F$  е **пълно**, ако  $[F] = P_2$ , т. е. ако всяка двоична функция се реализира с формула над  $F$ .

Едно тривиално пълно множество е самото  $P_2$ . Пример за непълно множество е  $\bar{x}$ , тъй като неговата обвивка се състои от тавтологичните функции  $x_1, x_2, x_3, \dots$  и отрицанията  $\bar{x}_1, \bar{x}_2, \bar{x}_3, \dots$

В следващия параграф ще покажем, че съществуват нетривиални пълни множества, различни от  $P_2$ .