



Логически изрази

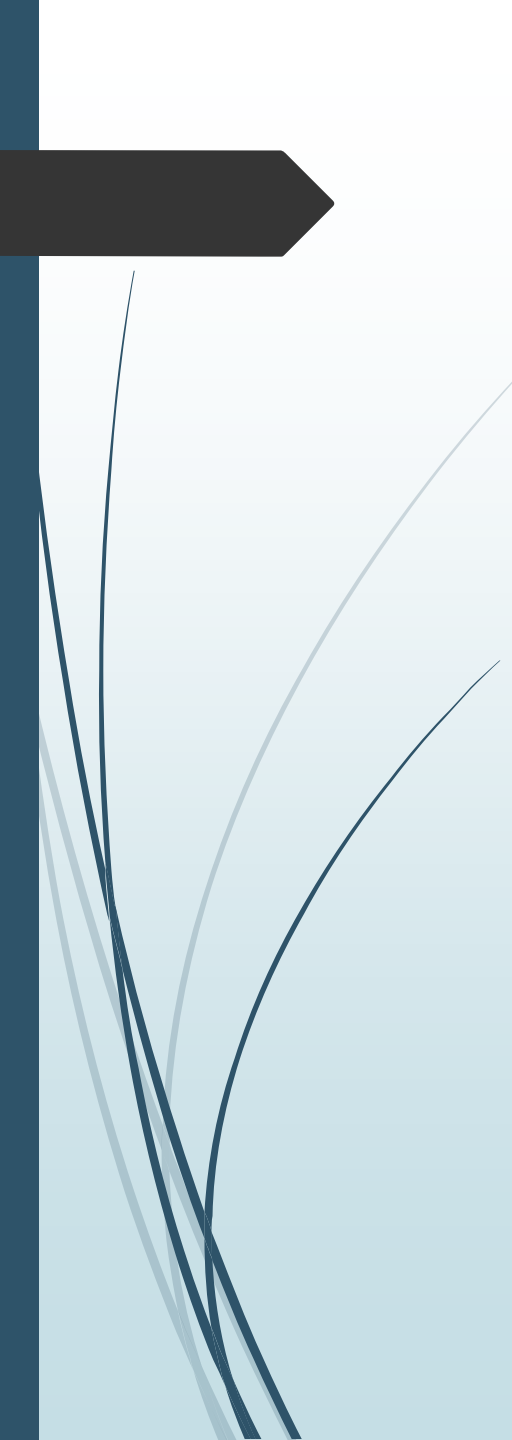
ОСНОВНИ ПОНЯТИЯ

- **Програма** – представлява последователност от инструкции, които след като бъдат декодирани от компютър, водят до решаване от страна на компютъра на дадена задача. Казано по друг начин – алгоритъм, написан на език за програмиране.
- **Алгоритъм** – всяка точно описана последователност от елементарни действия (указания), които се изпълняват, с цел решаването на конкретна задача от разглеждан клас задачи. Извършването на елементарно действие от изпълнителя се нарича стъпка. Всеки алгоритъм се състои от краен брой указания. Отличителните черти на алгоритмите са:
 - масовост – алгоритъмът се прилага върху всеки конкретен представител на определен кръг задачи;
 - дискретност (стъпковост) – всеки алгоритъм е съставен от стъпки, някои от които се изпълняват многократно с определена цикличност;
 - детерминираност (определеност) – стъпките в алгоритъма са строго определени;
 - резултатност – решението на задачата се получава след прилагането на краен брой стъпки на алгоритъма.

Видове алгоритми

- **Линейни** – съставени са от елементарни действия, които се изпълняват последователно;
- **Разклонени** – съдържат действия, които в зависимост от изпълнението на някакво условие, определят кои са следващите за изпълнение указания; позволяват изпълнението на алгоритъма да се управлява в зависимост от получените до момента резултати;
- **Циклични** – съдържат група от елементарни действия, които се изпълняват многократно. Дават възможност с малък брой команди да се представя голяма по обем еднотипна обработка на данни.

Обикновено алгоритмите имат смесен характер, като включват както редици от последователно изпълнявани команди, така също и разклонения и цикли.

- 
- **Програмен (софтуерен) продукт** – програма или съвкупност от взаимодействащи си програми, записани върху технически носител и придружени от съответната документация.
 - **Жизнен цикъл на програмен продукт** – обхваща целия период на неговото създаване и използване. За начало на ЖЦ се счита момента на възникване на идеята за създаването му. За край – преустановяването на използването на последното копие на програмния продукт. Могат да се разграничат физически и логически край.

- Процесът на разработване на програми е съвкупност от дейности, подредени във времето, чиято цел е създаване на коректни и ефективни програми.
- Етапи при разработване на програмите:
 1. Анализ на предметната област
 2. Проектиране на програмите (алгоритмизация)
 3. Проверка (верификация)
 4. Кодиране (реализация програмата на избран програмен език)
 5. Тестване

Синтаксис и семантика

6

- Езиците за програмиране са формални езици за описание на данните и алгоритмите за тяхната обработка посредством компютър. Те притежават следните четири елемента:
 - азбука – набор от разрешените за езика символи;
 - лексема (дума) – най-малката смислова единица на езика;
 - лексика – множеството от правила за изграждане на лексемите в езика (на синтактично правилни конструкции);
 - граматика – правилата, изграждащи синтактичните и семантичните единици на езика.
- **Синтаксисът** на един език включва правилата за построяване на граматически правилни изречения при употреба на езика. Всяка програма е поредица от допустими за езика символи, но не всяка поредица от символи е програма.

Синтаксис и семантика

7

- **Семантиката** включва правилата за построяване на смислово правилни изречения при боравенето с езика. Това е множество от правила, които дават възможност за еднозначно тълкуване на образуваните конструкции или това са правилата, по които се определя как се изпълняват програмите в даден език за програмиране.

Синтаксисът определя формата, а **семантиката** смисъла на една програма, записана на език от високо ниво.

- Семантика – смислово интерпретиране на езиковите конструкции;
 - влагане на формален смисъл в програмите;
 - специфицира действията на програмите.

Семантиката е насочена към значението на синтактично коректни програми!

Пример:

$z:=x; x:=y; y:=z;$

- колко са командите?
- какъв е смисълът (значението) на програмата?


Области на приложение на семантиката

- доказване правилността на програмите;
- методи за синтез;
- разработване на компилатори;
- разработване на нови езици за програмиране;
- сравнение на развитието на съществуващите езици за програмиране;
- програмиране на паралелни процеси;
- разработване на системи за работа в реално време;
- функционално програмиране.
- и др.

Логически функции

Съвременните компютри обработват числова, текстова, графична, звукова и видео информация. Всички тези видове информация в компютъра се представят чрез т.н. **двоичен код**. Използва се набор (азбука) от два символа – 1 и 0. Такова кодиране се нарича двоично. Всяка цифра от двоичния код съдържа в себе си 1 бит информация.

Изходната позиция в теорията на логическите функции е: **в резултат на изпълнението на произволна аритметична операция над двоични числа се получава ново двоично число.**




Двоична функция е тази двоична променлива, стойността, на която зависи от други двоични променливи. Тези зависимости са систематизирани в таблици за истинност. Тезата е известна като Булева алгебра по името на английския математик Джон Бул, който поставя основите на двузначната алгебра.

Съждение – изречение, за което може да се зададе въпрос дали е вярно или невярно.

Специален дял от науката логика, който изучава съжденията се нарича двузначна логика.

Булевите функции се представят чрез изрази от двоични променливи и знаци за логически операции.



Логическите операции биват **едноместни** (прости) с участието на **един** аргумент и **многоместни** (сложни) с участието на **n** аргумента.

Булевата алгебра използва **три основни операции: И, ИЛИ, НЕ**, които позволяват да се извършва събиране, изваждане, умножение, деление и сравнение на символи и числа. По този начин Бул описва подробно двоичната бройна система.

Логическа функция „Отрицание“

Логическата функция „Отрицание“ се реализира от схема „Не“, при която на изхода се получава инвертираната стойност на аргумента. Тя е едноместна логическа операция. Представя се с ключовата дума „NOT“. Отбелязва се със знак \neg преди аргумента или черта над него.

A	\bar{A}
0	1
1	0

Логическа функция „Дизюнкция“

Реализира се чрез двуместни логически операции събиране, затова и функцията е известна като логическо събиране. В програмните среди се представя с ключовата дума „OR“. Функцията се реализира по схемата „ИЛИ“.

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Броят на редовете в таблицата е равен на 2^n , където n е броят на аргументите. Следователно при $n=2$ таблицата има 4 реда.

Свойства на дизюнкцията

- асоциативност: $(x \vee y) \vee z = x \vee (y \vee z) = x \vee y \vee z$
- комутативност: $x \vee y = y \vee x$
- операции с 0: $x \vee 0 = x$
- операции с 1: $x \vee 1 = 1$
- поглъщане: $x \vee x = x$
- операции с отрицание: $x \vee \bar{x} = 1$

Логическа функция „Конюнкция“

Реализира се чрез двуместни логически операции умножение, затова и функцията е известна като логическо умножение. В програмните среди се представя с ключовата дума „AND“. Функцията се реализира по схемата „И“.

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Свойства на конюнкцията

- асоциативност: $(x \wedge y) \wedge z = x \wedge (y \wedge z) = x \wedge y \wedge z$
- комутативност: $x \wedge y = y \wedge x$
- операции с 0: $x \wedge 0 = 0$
- операции с 1: $x \wedge 1 = x$
- поглъщане: $x \wedge x = x$
- операции с отрицание: $x \wedge \bar{x} = 0$

„Сума по модул 2“ (изключващо ИЛИ, XOR)

Връща „истина“ само когато двата операнда имат различна стойност.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Свойства на „сума по модул 2“

\oplus

$$(x \oplus y) \oplus z = x \oplus (y \oplus z) = x \oplus y \oplus z$$

$$x \oplus y = y \oplus x$$

$$x \oplus 0 = x$$

$$x \oplus 1 = \bar{x}$$

$$x \oplus x = 0$$

$$x \oplus \bar{x} = 1$$

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

„Еквивалентност“

Връща „истина“ само когато операндите имат една и съща стойност. Свойство: $x \leftrightarrow y = \overline{x \oplus y} = (x \vee \bar{y}) \wedge (\bar{x} \vee y)$

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

„Стрелка на Пирс“ (NOR)

Връща „истина“ само когато и двата операнда са „лъжа“.

A	B	$A \downarrow B$
0	0	1
0	1	0
1	0	0
1	1	0

Свойства на стрелката на Пирс

$$(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$$

$$x \downarrow y = y \downarrow x$$

$$x \downarrow 0 = \bar{x}$$

$$x \downarrow 1 = 0$$

$$x \downarrow x = \bar{x}$$

$$x \downarrow \bar{x} = 0$$

$$x \downarrow y = \bar{x} \cdot \bar{y} = \overline{x \vee y}$$

„Щрих на Шефер“ (NAND)

Връща „лъжа“ само когато и двата операнда са „истина“.

A	B	A / B
0	0	1
0	1	1
1	0	1
1	1	0

Свойства на щриха на Шефер

$$(x / y) / z \neq x / (y / z)$$

$$x / y = y / x$$

$$x / 0 = 1$$

$$x / 1 = \bar{x}$$

$$x / x = \bar{x}$$

$$x / \bar{x} = 1$$

$$x / y = \overline{x \cdot y} = \bar{x} \vee \bar{y}$$

„Импликация“ (следствие)

Импликацията се чете “следва” или “ако ... , то ...”. Има два аргумента, като първият се нарича предпоставка, а вторият – следствие. Връща „лъжа“, само когато предпоставката „истина“, а следствието е „лъжа“. В останалите случаи импликацията има стойност „истина“. Ако е изпълнено $A \rightarrow B$, то логическият израз B е по-слаб от A .

Свойство: $x \rightarrow y = \bar{x} \vee y$

A	B	$A \rightarrow B$	$B \rightarrow A$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	1

Логически закони за еквивалентност

26

■ Закони за комутативност:

$$(A \wedge B) = (B \wedge A)$$

$$(A \vee B) = (B \vee A)$$

$$(A = B) = (B = A)$$

■ Закони за асоциативност:

$$A \wedge (B \wedge C) = (A \wedge B) \wedge C = A \wedge B \wedge C$$

$$A \vee (B \vee C) = (A \vee B) \vee C = A \vee B \vee C$$

■ Закони за дистрибутивност:

$$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$$

$$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$$

■ Закон на Морган:

$$\neg (A \wedge B) = \neg A \vee \neg B$$

$$\neg (A \vee B) = \neg A \wedge \neg B$$

- Закон на отрицанието:

$$\neg(\neg A) = A$$

- Закон за изключване на третия:

$$A \vee \neg A = T$$

- Закон за противоречието:

$$A \wedge \neg A = F$$

- Закон за импликацията:

$$(A \rightarrow B) = \neg A \vee B$$

- Закон за равенството:

$$(A = B) = (A \rightarrow B) \wedge (B \rightarrow A)$$

► Закон за опростяване \vee :

$$A \vee A = A$$

$$A \vee T = T$$

$$A \vee F = A$$

$$A \vee (A \wedge B) = A$$

► Закон за опростяване \wedge :

$$A \wedge A = A$$

$$A \wedge T = A$$

$$A \wedge F = F$$

$$A \wedge (A \vee B) = A$$

► Закон за тъждество:

$$A = A$$

Всяка булева функция може да се изрази чрез трите логически функции: конюнкция, дизюнкция и отрицание:

$$X \oplus Y = X \wedge \neg Y \vee \neg X \wedge Y$$

$$X \downarrow Y = \neg X \wedge \neg Y$$

$$X \leftrightarrow Y = X \wedge Y \vee \neg X \wedge \neg Y$$

$$X \leftarrow Y = X \vee \neg Y$$

$$X \rightarrow Y = \neg X \vee Y$$

$$X / Y = \neg X \vee \neg Y$$

Задачи:

30

Като се използват законите за еквивалентност, да се докаже следното:

$$(B \rightarrow C) = (\neg C \rightarrow \neg B)$$

Опростете логическите изрази:

■ $(b \vee a) \wedge (b \Rightarrow a) \wedge (a \Rightarrow b)$ Отг. $b \wedge a$

■ $(p \wedge (p \Rightarrow q)) \Rightarrow q$ Отг. T

■ $x \vee (\neg x \wedge y) \vee (x \wedge z) \vee (x \wedge \neg z)$

Решение:

$$= x \vee (\neg x \wedge y) \vee [x \wedge (z \vee \neg z)] =$$

$$= x \vee (\neg x \wedge y) \vee (x \wedge 1)$$

$$= x \vee (\neg x \wedge y) \vee x =$$

$$= x \vee x \vee (\neg x \wedge y)$$

$$= x \vee (\neg x \wedge y) =$$

$$= (x \vee \neg x) \wedge (x \vee y)$$

$$= 1 \wedge (x \vee y)$$

$$= x \vee y$$