

Дискретна математика

проф. д-р Тодорка Глушкова,
Катедра „Компютърни технологии“, ФМИ

5

Логика

Съдържание

- Съждителна логика
- Синтаксис
- Семантика
- Теория на доказателствата
- Предикатна логика
- Приложения

Логика

- Математиката, за разлика от други научни дисциплини работи само с неоспорима истина.
- Една от основните цели на математиците е да определят кои твърдения са верни и кои -не.
- Изследването на съждения, начина им на конструиране, смисъла и доколко може да се докаже верността им е предмет на дисциплината "математическа логика"
- Логиката е научна дисциплина. Корените и можем да търсим в древна Гърция, а първия опит да се постави на научна основа принадлежи на Аристотел в книгата му "Аналитика".

Логика

- Логиката е необходимо средство и за математици и за компютърни специалисти. С нейна помощ един математик може да докаже верността на едно твърдение, а един компютърен специалист - че една програма работи така както се очаква за всички възможни въвеждания на данни.
- **Логически оператори** - така както операциите и релациите в математиката се използват за съставяне на по-сложни изрази, така и логическите оператори ще ни помогнат да образуваме съставни твърдения.

Логика

- Числовата алгебра използва за означаване на операции и релации символи (+, -, <, > ...). Логиката също използва подобни символи, които съставляват нейния език.
- Алгебрата има свои закони - асоциативен, комутативен, дистрибутивен и др.
- Логиката има също свои закони, подобни на алгебричните.
- Съжителната логика, разглеждана като аналог на алгебрата се нарича "Логическа алгебра" или "Съжително смятане".

Съждителна логика

- Най-простият вариант на математическата логика
 - Общо подмножество (най-малък общ делител) на всички останали логики.
- Предмет: атомарни съждения и връзките между тях
- Предимства:
 - Просто изграждане
 - Ясна структура
- Недостатък:
 - Недостатъчна изразителна сила
- Въпреки това, значението ѝ е огромно
 - Теоретична основа за разработване на хардуер и софтуер
 - Поведението на комбинаторни хардуерни схеми на логическо ниво може да се представи директно като логически формули

Логически съюзи и съставни съждения

Съждението е твърдение, за което със сигурност можем да кажем дали е вярно или не. От синтактична гледна точка то е просто изречение с подлог и сказуемо.

Напр.: "Иван е човек", " $5 < 8$ " и т.н.

- Ако са дадени две съждения можем да ги свържем като използваме различни съюзи в съставно съждение.

Логически съюзи и съставни съждения

Съюзите за връзка могат да бъдат: "И", "ИЛИ", "АКО...ТО".

- Напр.: За простите съждения- " $4 < 7$ "; "8 е четно число", можем да образуваме съставните съждения:
- " $4 < 7$ и 8 е четно число";
- " $4 < 7$ или 8 е четно число";
- "Ако $4 < 7$, то 8 е четно число".

За всяко едно от получените съставни съждения можем да кажем дали е вярно или не, т.е. отново получаваме съждение

Следователно тези съюзи са затворени оператори в множеството на всички съждения

Логически съюзи и съставни съждения

За всяко съждение можем да получим неговото отрицание с помощта на частицата "не".

- Пример: "4 не е < 7 "

Отрицанието е унарна логическа операция.

- **Въпрос:** Как можем да установим верността на съставните съждения?
- **Отговор:** С верностна таблица!

Логически оператори

- Нека е дадено съждението P . Тогава **отрицанието** на P ще записваме $\neg P$ и ще казваме, че $\neg P$ е вярно, когато P е невярно и обратно.
- Верностната таблица на отрицанието е:

P	$\neg P$
T	F
F	T

T – истина

F – лъжа

Логически оператори

- Пример: $P = "4 < 7"$ е вярно, защото $\neg P = "4 \text{ не е } < 7"$ е грешно.
- Забележка: Отрицанието има аналог в алгебрата – знакът "-".

КОНЮНКЦИЯ

- **Конюнкция**: Нека са дадени две съждения P и Q . Конюнкция на двете съждения $P \wedge Q$ е съждение, което е вярно само когато и P и Q са верни.
- Верностната таблица:

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

КОНЮНКЦИЯ

- Конюнкцията има за аналог в говоримия език съюза “и”, а в алгебрата – умножението, поради което се нарича още логическо умножение.
- Пример: Съждението “ $(4 < 7) \wedge (8 \text{ е четно число})$ ” е вярно съгласно първи ред от верностната таблица.

ДИСЮНКЦИЯ

- **Дисюнкция:** Нека са дадени две съждения P и Q . Дисюнкцията на тези две съждения $P \vee Q$ е съждение, което е вярно, когато поне едно от двете дадени съждения е вярно и грешно, ако и двете са грешни.
- Верностната таблица:

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

ДИСЮНКЦИЯ

- Дисюнкцията има за аналог в говоримия език съюза "или", а в алгебрата – събирането, поради което се нарича още логическо събиране.
- Пример: " $(4 < 7) \vee (8 \text{ е не четно число})$ " е вярно поради втори ред на верностната таблица.
- След като дефинирахме тези три логически оператора, можем да ги прилагаме в различни комбинации и да получаваме съставни съждения. Тяхната верностна стойност можем да получим чрез верностна таблица.
- Например: Ако P, Q и R са съждения да получим верностните стойности на съставното съждение: $R \wedge \neg (P \wedge Q)$.

P	Q	R	$P \wedge Q$	$\neg(P \wedge Q)$	$R \wedge \neg(P \wedge Q)$
T	T	T	T	F	F
T	T	F	T	F	F
T	F	T	F	T	T
T	F	F	F	T	F
F	T	T	F	T	T
F	T	F	F	T	F
F	F	T	F	T	T
F	F	F	F	T	F

Импликация

- **Импликация:** Ако P и Q са две съждения импликация $P \rightarrow Q$ наричаме съждението, което е грешно, само когато P е вярно, а Q - невярно и вярно във всички останали случаи.
- Ще наричаме P -хипотеза, а Q - заключение.

В говоримия език импликацията се замества с "ако...тогава".

В примера: " $(4 < 7) \rightarrow (8 \text{ е четно число})$ " е вярно, съгласно първи ред от верностната таблица.

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Импликация

- В разговорния език можем да го кажем и по друг начин:
 - ✓ "Ако P , то Q ";
 - ✓ " P само ако Q ";
 - ✓ " P е достатъчно условие за Q ";
 - ✓ " Q е необходимо условие за P ".
- **Пример:** P ="Вън вали."; Q ="На небето има облаци." Тогава:
 - ✓ "Ако вън вали, на небето има облаци."
 - ✓ "Вън вали, само ако на небето има облаци."
 - ✓ Вън да вали е достатъчно условие за това на небето да има облаци.
 - ✓ На небето да има облаци е необходимо условие вън да вали.

Импликация

- Ще отбележим, че ако Q е вярно съждение, импликацията $P \rightarrow Q$ е винаги истина.

(напр. $X < 7 \rightarrow 2 + 2 = 4$ е вярно независимо от стойността на X .)

- Също, ако при дисюнкцията и конюнкцията реда на записване на съжденията е без значение ($P \wedge Q, Q \vee P$), то при импликацията той е много важен. Съждението $P \rightarrow Q$ не е еквивалентно на $Q \rightarrow P$.
- В горния пример съждението "Ако вълн вали, то на небето има облаци" не е еквивалентно на "Ако на небето има облаци, то вълн вали", тъй като денят може да е просто облачен.
- Когато, обаче, и двете съждения $P \rightarrow Q$ и $Q \rightarrow P$ са едновременно верни, тогава казваме, че съжденията P и Q са еквивалентни и записваме $P \leftrightarrow Q$.

Еквиваленция

- **Еквиваленцията** $P \leftrightarrow Q$ е вярна, когато P и Q получават еднакви верностни стойности и невярна в останалите случаи.
- Верностна таблица:

P	Q	$P \leftrightarrow Q$
T	T	T
T	F	F
F	T	F
F	F	T

Еквиваленция

- В математиката еквивалентността се изговаря като "тогава и само тогава, когато" или като "Р е необходимо и достатъчно условие за Q".
- **Например:** Триъгълникът е правоъгълен тогава и само тогава, когато сборът от квадратите на катетите е равен на квадрата на хипотенузата.

Изключващо „Или“- XOR

- **Изключващото „Или“** $P \oplus Q$ е вярна, само когато P и Q получават различни верностни стойности и невярна в останалите случаи.
- Верностна таблица: Нарича се още „сума по модул 2“ или „изключващо ИЛИ“.
- Означава се с:

$$f = P \oplus Q$$

От верностната таблица се вижда, че XOR е противоположен на еквиваленцията и затова понякога се нарича антиеквиваленция и се бележи с \Leftrightarrow

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

Побитови операции или Шифри срещу Кодове

- За повечето хора това е все едно да ги попиташ каква е разликата между смесване и разбъркване.
- Кодът е свързване на някаква смислена единица – като дума, изречение или фраза – с нещо друго – обикновено по-кратка група символи. Например можем да направим код, в който думата "apple" (ябълка) е записана като 67.
- Кодовата книга е списък от тези връзки.
- За да създаваме и разчитаме кодове се нуждаем от кодова книга или кодова формула.
- Тогава какво е шифър?

Побитови операции или Шифри срещу Кодове

- Най-важното е, че шифрите не включват значение.
- Те са механични операции, познати като алгоритми, които се прилагат върху отделни букви или малки групи от тях.
- Например в Цезаровия шифър всяка буква от азбуката се превръща в друга буква. Например $A \rightarrow D$, $B \rightarrow E$ и $C \rightarrow F$, според някакво изместване, в този случай четири.
- Този вид шифър е познат като **шифър с отместване**.
- В този случай не ни трябва кодова книга.

Побитови операции или Шифри срещу Кодове

- Вместо това следваме поредица от инструкции, познати още като **алгоритъм**, където преместваме всяка буква с някакъв брой позиции.
- Алгоритъмът изисква някаква споделена информация, позната като **ключ**.
- В горния пример, в който $A \rightarrow D$, ключът е четири. Този споделен ключ се изисква от двете страни, за да **кодират** съобщения: $HELLO \rightarrow KHOOR$, както и да **декодират** съобщения: $KHOOR \rightarrow HELLO$.

Побитови операции или Шифри срещу Кодове

- Да се върнем на въпроса: Каква е разликата между код и шифър?
- Кодовете се отнасят към **семантичното** значение, докато шифрите оперират със **синтаксиса**, символите.
- Кодът се запазва в кодова книга, докато шифрите превръщат отделните символи един в друг според някакъв алгоритъм.

Побитови операции или **Шифри срещу Кодове**

- Най-добрият шифър с отместване е шифърът с еднократен код, при който всеки символ се шифрова с различно случайно отместване.
- Той се осъществява чрез прилагането на последователност от случайни замествания с дължина равна на дължината на съобщението.
- Този шифър е трудно пробиваем, тъй като всяка комбинация е равновероятна на останалите, а броят комбинации нараства експоненциално.

Побитови операции или Шифри срещу Кодове

- Например при шифроване на думата КОД с възможно отместване 30 (колкото са буквите в българския език) получаваме

К **О** **Д**

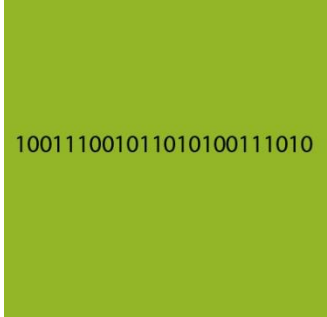
30* 30* 30=27000 различни
равновероятни трибуквени комбинации.

Побитови операции или Шифри срещу Кодове

- Причината за това е използването на побитовите побитовите операции **AND** (и), **OR** (или) и **XOR** (изключващо или).
- И най-вече, трябва да разберем защо XOR трябва да се използва, когато се използва шифър с еднократен код на компютри.
- **Побитово** просто означава, че работим с отделни битове или **двоични числа**.
- Във всяка съвременна компютъризирана схема за критиране ние представяме символите с 0 и 1.

Криптиране на цветове

- Всеки RGB цвят е комбинация от три осембитови числа. Например: (R) = 156, (G) = 161 и (B) = 58.
- Ако изразим числата двоично, получаваме: R=10011100, G=10110101, B=00111010.
- Можем да ги представим заедно така:
100111001011010100111010
- Сега генерираме отместване с произволно число със същата дължина (напр. като хвърляме 24 пъти монета)
- За да извършим кодирането с еднократен случаен ключ, трябва да изберем правилната операция, така че получената последователност да може да е който и да е цвят с еднаква вероятност.
- Нека да разгледаме различните операции: **AND, OR, XOR.**



100111001011010100111010

Криптиране на цветове

Логическо «И» (AND)

неговата таблица на истинност:

0 AND 0 = 0

1 AND 1 = 0

2 AND 0 = 0

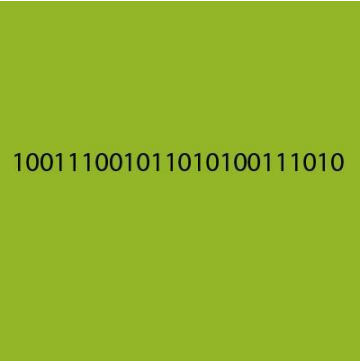
1 AND 1 = 1

Нека опитаме:

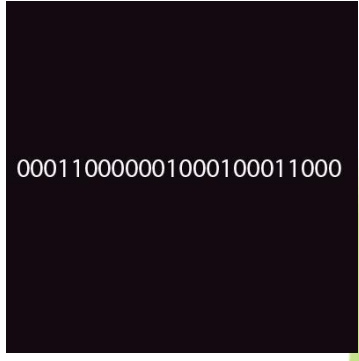
**100111001011010100111010 AND 010110100001101111011000 =
000110000001000100011000**

Полученият цвят е много тъмно лилаво.

Когато прилагаме операцията **AND**, резултът **не може да е по-голям от него.**



100111001011010100111010



000110000001000100011000

Криптиране на цветове

Логическо «Или» (OR)

0 OR 0 = 0

0 OR 1 = 1

1 OR 0 = 1

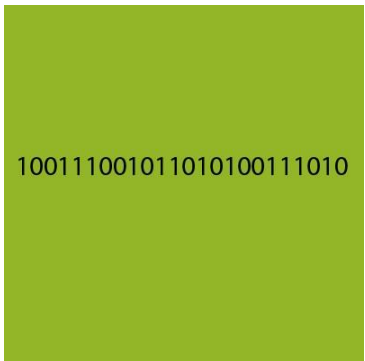
1 OR 1 = 1

Нека опитаме:

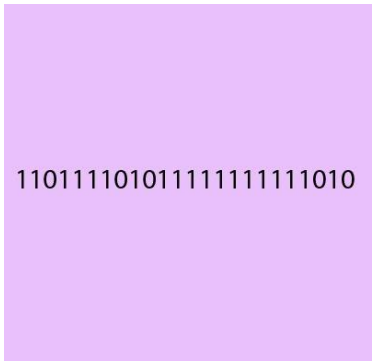
100111001011010100111010 **OR** 010110100001101111011000
= 11011110101111111111010

Резултатът е светло лилаво.

Когато прилагаме операцията **OR**, получената последователност **не може да е по-малка**.



100111001011010100111010



11011110101111111111010

Криптиране на цветове

Изключващо «Или» (XOR)

0 XOR 0 = 0

1 XOR 1 = 1

2 XOR 0 = 1

1 XOR 1 = 0

Нека опитаме:

100111001011010100111010 **XOR** 010110100001101111011000
= 110001101010111011100010

- Резултатът е малко по-тъмно лилаво в сравнение с OR.
- Когато използваме операцията XOR, **получената последователност може да бъде всяка възможна**
- Ако вземем някакъв криптиран цвят, всичко което знаем е, че първоначалният цвят "е еднакво вероятно да бъде всеки цвят" и е необходимо сляпо отгатване (1/16 милиона).

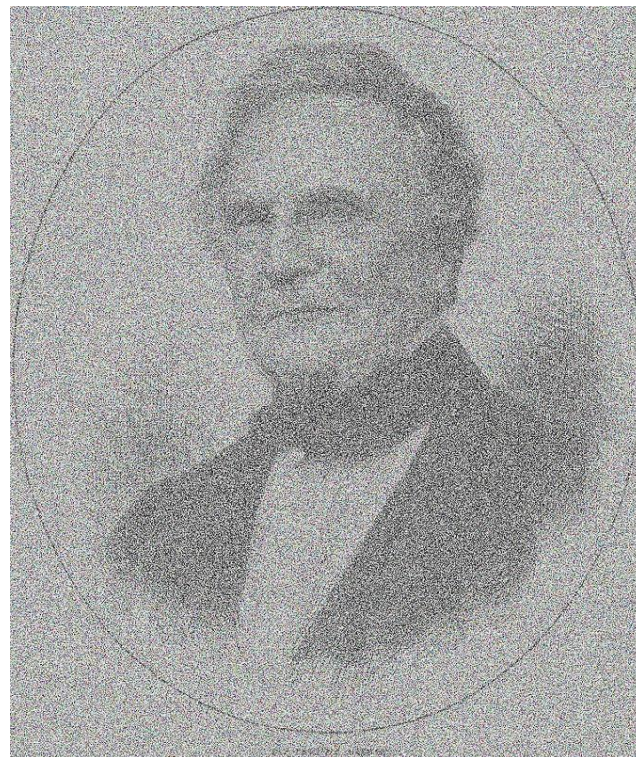
100111001011010100111010

110001101010111011100010

Попикселно криптиране на изображения



AND



Попикселно криптиране на изображения



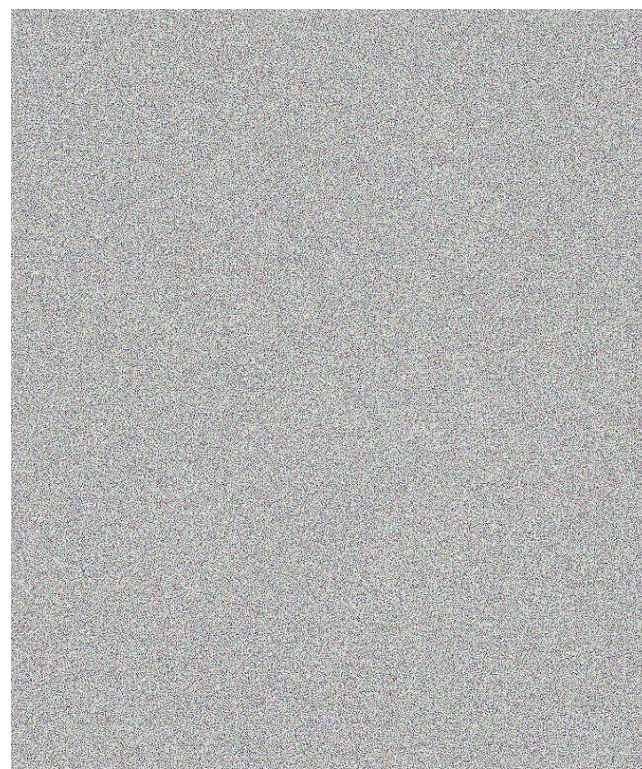
OR



Попикселно криптиране на изображения



XOR



Тавтология и еквивалентност

Някои съставни съждения са винаги верни, а други – винаги неверни независимо от верностните стойности на съставляващите ги съждения.

- **Например:** $P \vee \neg P$ е винаги вярно твърдение, докато $P \wedge \neg P$ е винаги невярно.
- Това можем да видим от верностните им таблици:

P	$\neg P$	$P \vee \neg P$
T	F	T
F	T	T

P	$\neg P$	$P \wedge \neg P$
T	F	F
F	T	F

Тавтология и противоречие

- **Тавтология**: Всяко съждение, което е винаги вярно, независимо от верностните стойности на съставляващите го съждения.
- **Противоречие**: Съждение, което е винаги невярно. (Второто съждение от горния пример.)
- Можем лесно да ги разпознаем, ако във верностната таблица получим само Т (тавтология) или само F (противоречие).

Пример: Да докажем, че $\neg(P \vee Q) \rightarrow \neg P \wedge \neg Q$ е тавтология.

P	Q	$\neg P$	$\neg Q$	$P \vee Q$	$\neg P \wedge \neg Q$	$\neg(P \vee Q)$	$\neg(P \vee Q) \rightarrow \neg P \wedge \neg Q$
T	T	F	F	T	F	F	T
T	F	F	T	T	F	F	T
F	T	T	F	T	F	F	T
F	F	T	T	F	T	T	T

Тавтология и еквивалентност

- Нека S_1 и S_2 са две съждения. Казваме, че те са еквивалентни, когато двете колони във верностната таблица, в които те получават стойностите си са еднакви.
- В примера това са колоните за $\neg P \wedge \neg Q$ и $\neg (P \vee Q)$. Тогава можем да кажем, че:

$$\neg P \wedge \neg Q \Leftrightarrow \neg (P \vee Q)$$

е еквивалентност.

Логически твърдения

- В математическата логика съществуват два начина за доказване на логически еквивалентности:
 - дедуктивен
 - манипулативен (с верностна таблица).
- Често манипулативния подход води до бързи резултати, но при верностни таблици с много логически променливи, този подход е практически неприложим.
- Затова ще въведем някои базови еквивалентности, с чиято помощ чрез дедуктивни методи ще можем да докажем всяка друга еквивалентност.

ТЕОРЕМА 1. Ако P, Q и R са твърдения, T е вярно твърдение, а F е грешно, следните двойки твърдения са логически еквивалентни:

1) Комутативни закони:

$$P \vee Q \Leftrightarrow Q \vee P$$

$$P \wedge Q \Leftrightarrow Q \wedge P$$

2) Асоциативни закони:

$$P \vee (Q \vee R) \Leftrightarrow (P \vee Q) \vee R$$

$$P \wedge (Q \wedge R) \Leftrightarrow (P \wedge Q) \wedge R$$

3) Дистрибутивни закони:

$$P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$$

$$P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$$

4) Комплиментарни закони:

$$P \vee \neg P \Leftrightarrow T$$

$$P \wedge \neg P \Leftrightarrow F$$

5) Зако́ни на идентитета:

$$P \wedge T \Leftrightarrow P$$

$$P \vee F \Leftrightarrow P$$

6) Доминантни зако́ни:

$$P \vee T \Leftrightarrow T$$

$$P \wedge F \Leftrightarrow F$$

7) за иде́мпотентност:

$$P \wedge P \Leftrightarrow P$$

$$P \vee P \Leftrightarrow P$$

8) Зако́н за двойното отрицание:

$$\neg(\neg P) = P$$

9) Зако́ни на Де Морган:

$$\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$$

$$\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$$

10) Закон за импликацията: $P \rightarrow Q \Leftrightarrow \neg P \vee Q$

11) Закон за отрицание на импликацията:

$$\neg(P \rightarrow Q) \Leftrightarrow \neg P \wedge Q$$

12) Закон за контрапозицията:

$$P \rightarrow Q \Leftrightarrow \neg Q \rightarrow \neg P$$

13) Закон за еквиваленцията:

$$P \leftrightarrow Q \Leftrightarrow (P \rightarrow Q) \wedge (Q \rightarrow P)$$

Логически изводи

- Логическите изводи ни позволяват да конструираме коректни доказателства.
- Дефиниция: Нека $S1$ и $S2$ са съставни съждения. Казваме, че $S2$ следва от $S1$, т.е. $S1 \Rightarrow S2$, ако за всяко разпределение на верностните стойности на съждителните променливи в $S1$ и $S2$, от верността на $S1$ следва верността на $S2$.

Пример

- **Пример:** За да докажем, че: $P \wedge (P \rightarrow Q) \Rightarrow Q$,
Конструираме верностната таблица и сравняваме колоната $P \wedge (P \rightarrow Q)$ с колона Q .

P	Q	$P \rightarrow Q$	$P \wedge (P \rightarrow Q)$
T	T	T	T
T	F	F	F
F	T	T	F
F	F	T	F

Логически изводи

Теорема 2 Следващите логически изводи са верни. P, Q, R са някакви съждения, а T и F са вярно и невярно съждение:

- 1) $(P \rightarrow Q) \wedge P \Rightarrow Q$
- 2) $P \Rightarrow P \vee Q$
- 3) $P \wedge Q \Rightarrow P$
- 4) $\top \rightarrow P \Rightarrow P$
- 5) $\top \rightarrow F \Rightarrow P$
- 6) $F \Rightarrow P$
- 7) $P \Rightarrow T$
- 8) $(P \rightarrow Q) \wedge (Q \rightarrow R) \Rightarrow P \rightarrow R$

Импликация и логически извод

- Да отбележим връзката между импликацията (\rightarrow) и логическия извод (\Rightarrow); между еквиваленцията (\leftrightarrow) и логическата еквивалентност (\Leftrightarrow).
- **Теорема 3.** Нека P и Q са съждения. Тогава:
 - 1) $P \Rightarrow Q$ тогава и само тогава, когато $P \rightarrow Q$;
 - 2) $P \Leftrightarrow Q$ тогава и само тогава, когато $P \leftrightarrow Q$;

Използвана литература в курса

- D. W. Hoffmann, Theoretische Informatik, Hansen Verlag, 2009
- H. P. Gumm, M. Sommer, Einfuehrung in die Informatik, Oldenbourg Wissenschaftsverlag, 2004
- J. W. Grossman, Discrete Mathematics, Macmillan Pub. Co., 1990
- К. Манев, Увод в дискретната математика, КЛМН, 2005
- Й. Денев, Р. Павлов, Я. Демирович. Дискретна математика. Наука и изкуство, София, 1984.

Използвана литература в курса

- Д. Байнов, С. Костадинов, Р. Павлов, Л. Луканова. Ръководство за решаване на задачи по дискретна математика. Университетско издателство "Паисий Хилендарски", Пловдив, 1990.
- В.А. Успенский, Машина Поста, Москва, Наука, 1988, ISBN 5-02-013735-9.
- L. Lovasz, J. Pelikan, K. Vesztergombi, Discrete Mathematics – Elementary and Beyond, Springer Verlag, New York, 2003, ISBN 0-387-95584-4.

Използвана литература в курса

- E. Bender, S. Williamson, A Short Course in Discrete Mathematics, Dover, 2006, ISBN 0-486-43946-1.
- P. Linz, An Introduction to Formal Languages and Automata, Jones and Bartlett Publishers, 6-th edition, Jones & Bartlett Publishers, ISBN-13: [9781284077247](#), 2016
- Kenneth H. Rosen, Kamala Krithivasan, Discrete mathematics and its application, McGraw-Hill Companies, 7-th edition, ISBN 978-0-07-338309-5, 2012

Използвана литература в курса

- Owen D. Byer, Deirdre L. Smeltzer, Kenneth L. Wantz, Journey into Discrete Mathematics, AMS, MAA Press, Providence Rhode Island, ISBN 9781470446963, 2018
- Christopher Rhoades, Introductory Discrete Mathematics, Willford Press, ISBN 1682854922, 9781682854921, 2018
- David Liben-Nowell, Discrete Mathematics for Computer Science, Wiley, 2017, ISBN 1119397197, 9781119397199, 2017.
- <http://www.jflap.org/> - софтуерна среда