МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ГРАЖДАНСКОЙ АВИАЦИИ

**А.В. Самохин, Ю.И. Дементьев**

# МАТЕМАТИЧЕСКАЯ ЛОГИКА

ПОСОБИЕ
по выполнению лабораторных работ

*для студентов I курса
направления 09.03.01
очной формы обучения*

**Москва-2016**

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА**
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ**
**БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**
**ВЫСШЕГО ОБРАЗОВАНИЯ**
**«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)**

**Кафедра высшей математики**
А.В. Самохин, Ю.И. Дементьев

# МАТЕМАТИЧЕСКАЯ ЛОГИКА

ПОСОБИЕ
по выполнению лабораторных работ

*для студентов I курса*
*направления 09.03.01*
*очной формы обучения*

**Москва-2016**

ББК 517
С17

Рецензент канд. физ.-мат. наук, доц. О.Г. Илларионова

Самохин А.В., Дементьев Ю.И.
С17 Математическая логика. Пособие по выполнению лабораторных работ. - М.: МГТУ ГА, 2016. - 12 с.

Данное пособие издаётся в соответствии с рабочей программой учебной дисциплины «Математическая логика» по учебному плану направления 09.03.01 для студентов I курса очной формы обучения.

Пособие содержит образцы выполнения лабораторных работ по дисциплине «Математическая логика» и необходимые для этого сведения по математической программе maple.

Рассмотрено и одобрено на заседаниях кафедры 14.09.2016 г. и методического совета 25.10.2016 г.

# Лабораторная работа №1
# Нормальные формы высказываний

```
> restart; with(Logic);
     [&and, &iff, &implies, &nand, &nor, &not, &or, &xor,
        BooleanSimplify, Canonicalize, Contradiction, Dual,
        Environment, Equivalent, Export, Implies, Import, Normalize,
        Random, Satisfy, Tautology, TruthTable]
```

Генерация случайных булевых функций

```
> a := Random([x, y], form = CNF)
                      a := x &or y

> d := Random({x, y, z}, form = MOD2)
                d := x y z + x y + x z + y z + x + y

> c := Random([x, y, z], form = DNF)
     c :=
        ((y &and z) &and &not(x) &or (y &and &not(x))
         &and &not(z)) &or (z &and &not(x)) &and &not(y)

> b := Random({x, y})
     b :=
        ((x &and y &or x &and &not(y)) &or y &and &not(x))
         &or &not(x) &and &not(y)
```

Преобразование стандартного Maple к булевой форме и многочлену Жегалкина

```
>Export(`&and`(a, b, c) &or b, form = boolean)
     (x or y) and (x and y or x and not y or y and not x or not (x
        or y)) and (y and z and not x or y and not x and not z or z
        and not x and not y) or x and y or x and not y or y and not
        x or not (x or y)

> Export((`&nor`(a, b, c)) &and b, form = boolean)
     not (x or y or x and y or x and not y or y and not x or not (x
        or y) or y and z and not x or y and not x and not z or z
        and not x and not y) and (x and y or x and not y or y and
        not x or not (x or y))
```

> Export(`&not`(a) &or b, form = MOD2)

$$1+(1+(x+1)\,(y+1))^2\,(x\,y+1)\,(x\,(y+1)+1)\,(y\,(x+1)+1)$$

Эквивалентность

> Equivalent(a &and (a &or b), a)

true

Таблицы истинности

> T1 := TruthTable(a &xor b, [x, y, z]); $T1_{true,\ false,\ true}$

T1 := table([(false, false, true) = true, (false, true, true) = false, (false, true, false) = false, (true, false, true) = false, (true, true, true) = false, (true, false, false) = false, (false, false, false) = true, (true, true, false) = false])

false

> T2 := TruthTable(`&not`(a) &nor b &iff c, [x, y, z]); $T2_{true,\ false,\ false}$

T2 := table([(false, false, true) = false, (false, true, true) = false, (false, true, false) = false, (true, false, true) = true, (true, true, true) = true, (true, false, false) = true, (false, false, false) = true, (true, true, false) = true])

true

> T3 := TruthTable(a &iff b, [x, y, z], form = MOD2); $T3_{1,\ 0,\ 1}$

T3 := table([(1, 1, 0) = 1, (0, 0, 1) = 0, (1, 0, 0) = 1, (0, 1, 0) = 1, (1, 1, 1) = 1, (0, 1, 1) = 1, (0, 0, 0) = 0, (1, 0, 1) = 1])

1

Преобразование к стандартному виду Maple

> Import(a or b or c, form = boolean)

((x &or y) &or (((x &and y &or x &and &not(y)) &or y &and &not(x)) &or &not(x) &and &not(y))) &or (((y &and z) &and &not(x) &or (y &and &not(x)) &and &not(z)) &or (z &and &not(x)) &and &not(y))

```
> Import(not (a xor b) ⇒ c, form = boolean)
        Logic:-&implies(&not((x &or y) &xor (((x &and y &or x
            &and &not(y)) &or y &and &not(x))
            &or &not(x) &and &not(y))),
            ((y &and z) &and &not(x) &or (y &and &not(x))
            &and &not(z)) &or (z &and &not(x)) &and &not(y))

> Import(x (y + 1) + z + 1, form = MOD2)
                        &not(x &and &not(y) &xor z)
```

Двойственное выражение

```
> Dual((a &and `&not`(a)) = false)
                    (x &and y &or &not(x &and y)) = true

> Dual(a &implies b)
        Logic:-&implies(&not(x &and y),
            &not((((x &or y) &and (x &or &not(y))) &and (y
            &or &not(x))) &and (&not(x) &or &not(y))))

> Dual(`&not`(a) &nor b &iff c)
        &not(x &and y) &nand ((((x &or y) &and (x &or &not(y)))
            &and (y &or &not(x))) &and (&not(x) &or &not(y)))
            &xor (((y &or z) &or &not(x)) &and ((y &or &not(x))
            &or &not(z))) &and ((z &or &not(x)) &or &not(y))
```

Тавтологии и противоречия

```
> Tautology((a &and b &or `&not`(a)) &or `&not`(b))
                                   true
> Tautology((a &iff b) &or b, 'p')
                                   true
> Tautology(a &or `&not`(a), 'p')
                                   true
> Contradiction((a &or `&not`(a &and b)) &nor b)
                                   true
> Contradiction((a &iff b) &or b, 'p')
                                   false
> Contradiction(`&not`(a) &and a, 'p')
                                   true
```

Преобразование к канонической форме

> Canonicalize(a &iff b, form = MOD2)

x y + x + y

> Canonicalize(a &iff b, {x, y, z}, form = CNF)

((x &or y) &or z) &and ((x &or y) &or &not(z))

> Canonicalize(a &iff b, {x, y, z}, form = DNF)

(((((x &and y) &and z &or (x &and y) &and &not(z)) &or (x
&and z) &and &not(y))
&or (x &and &not(y)) &and &not(z))
&or (y &and z) &and &not(x))
&or (y &and &not(x)) &and &not(z)

Упрощение выражений (Убираем лишние скобки и тавтологии)

Environment(2); (a &and b) &and ( a &or a)

(x &or y) &and (((x &and y &or x &and &not(y)) &or y
&and &not(x)) &or &not(x) &and &not(y))

BooleanSimplify(&not a &or a)

true

# Лабораторная работа №2
# Нечеткая логика и нечеткие множества

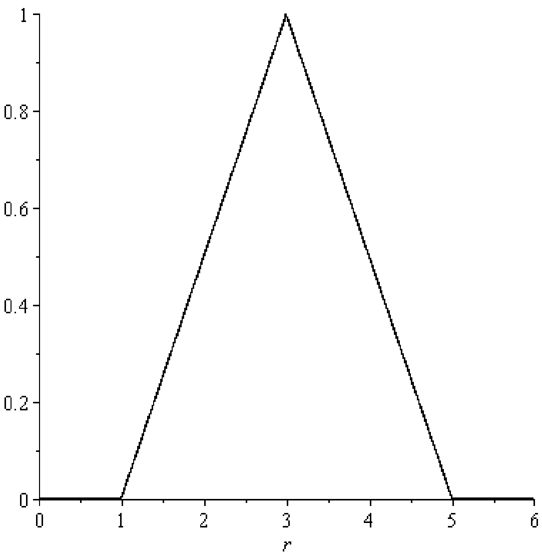
> restart; libname := libname, "F:/FuzzySets3" :
#libname нужен, если содержимое директории FuzzySets3 нельзя поместить в lib папку Maple

> with(FuzzySets[RealDomain])

$$[\text{Complement, Concentrate, Controller, Core, Cut, Defuzzify, Diffuse, Dilate, Equals, FuzzySet}, \Gamma, \text{Height, Intensify, L}, \Lambda, \text{Map, Map2, Normalize}, \Pi, \text{Partition, Support, implies, in, intersect, minus}, \mu, \text{plot, subset, union}]$$

6 конструкторов нечётких множеств на действительной прямой, Gamma, L, Lambda, Partition, and PI.

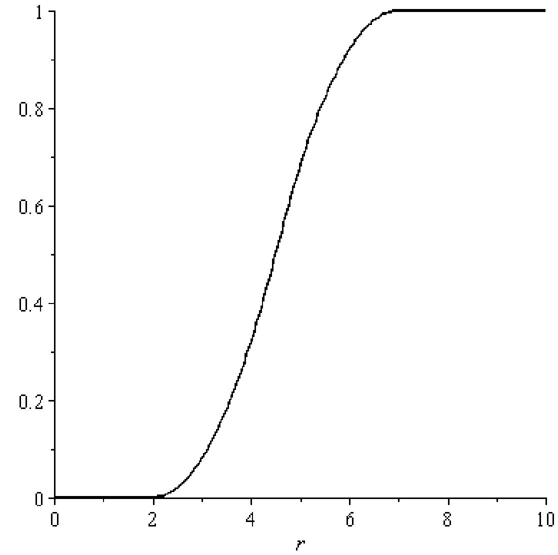
> Fz := Λ(1, 3, 5) :
> plot(Fz, 0 ..6)




> Gz := Gamma(2, 7, model = quadratic)

$$Gz := \begin{cases} 0 & r \le 2 \\ \frac{2}{25}(r-2)^2 & r < \frac{9}{2} \\ 1 - \frac{2}{25}(r-7)^2 & r < 7 \\ 1 & \text{otherwise} \end{cases}$$

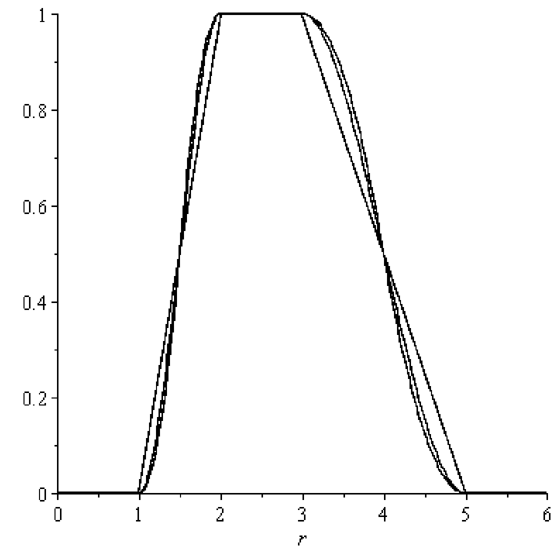
> plot(Gz, 0 ..10)




> Pz := Π(1, 2, 3, 5)

$$Pz := \begin{cases} 0 & r \leq 1 \\ r - 1 & r < 2 \\ 1 & r \leq 3 \\ -\frac{1}{2}\, r + \frac{5}{2} & r < 5 \\ 0 & \text{otherwise} \end{cases}$$

> plot([Π(1, 2, 3, 5), Π(1, 2, 3, 5, model = quadratic), Π(1, 2, 3, 5, model = cubic)], 0 ..6)



Степень принадлежности нечёткому множеству

> $4 \in Gz$

$$\frac{8}{25}$$

> $\pi \in Gz;\ \pi \in Pz$

$$\frac{2}{25}(\pi-2)^2$$

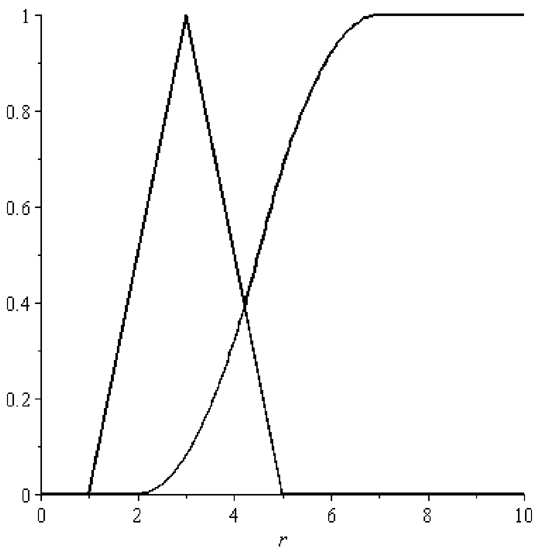
$$-\frac{1}{2}\pi+\frac{5}{2}$$

> evalf(%)

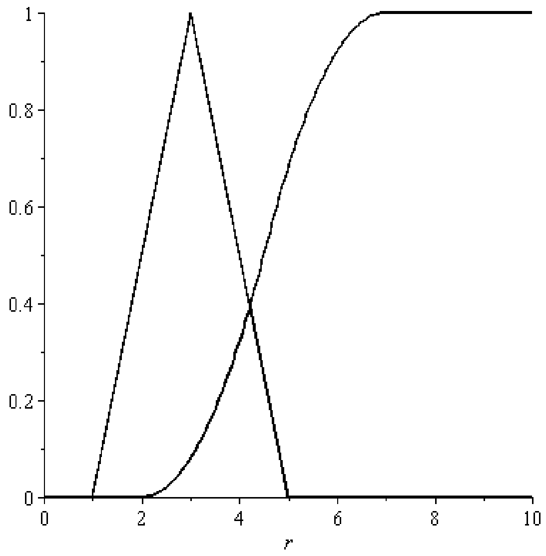
$$0.929203673$$

Можно составить unions, intersections, differences, и implicationsразных множеств:

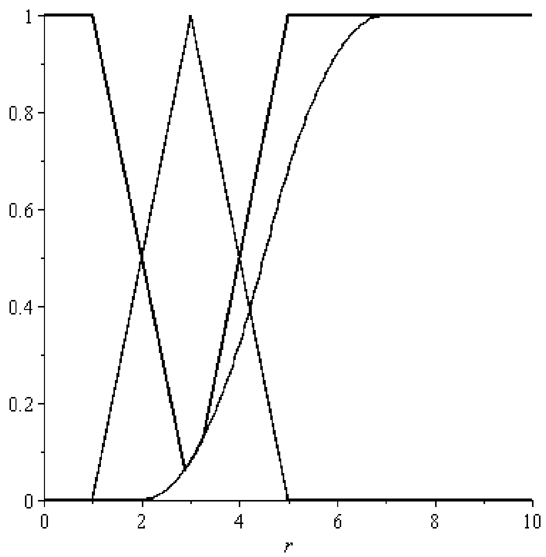
> plot([Fz, Gz, Fz $\cup$ Gz], 0 ..10, color = [red, blue, black],
thickness = [1, 1, 2])



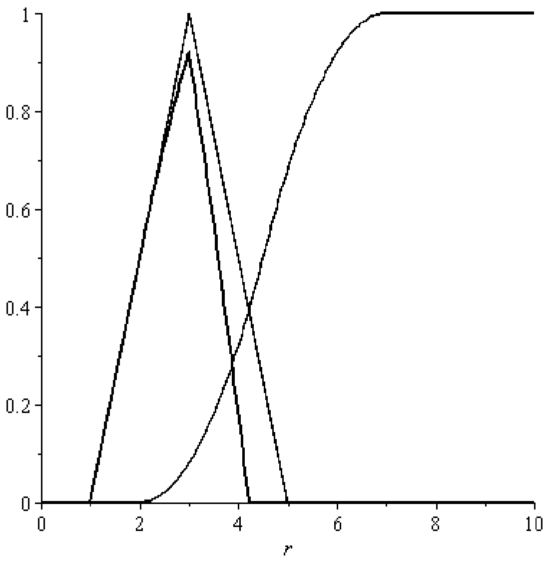
> plot([Fz, Gz, Fz ∩ Gz], 0 ..10, color = [red, blue, black], thickness = [1, 1, 2])




> plot([Fz, Gz, Fz ⇒ Gz], 0 ..10, color = [red, blue, black], thickness = [1, 1, 2])



```
> plot([Fz, Gz, Fz\Gz], 0..10, color = [red, blue, black], thickness
      = [1, 1, 2])
```



Пользуясь определениями and, or, not, и implies, для пары множеств S и T, определим функции принадлежности $S \cup T$ , $S \cap T$ и $\neg S$ (дополнения к S) следующим образом:

$$\mu_{(S\ \cup\ T)}(x) = \left(\mu_S(x)\ \text{ or }\ \mu_T(x)\right) = \max\left(\mu_S(x), \mu_T(x)\right)$$
$$\mu_{(S\ \cap\ T)}(x) = \left(\mu_S(x)\ \text{ and }\ \mu_T(x)\right) = \min\left(\mu_S(x), \mu_T(x)\right)$$
$$\mu_{\neg S}(x) = \left(\text{not }\ \mu_S(x)\right) = 1 - \mu_S(x)$$
$$\mu_{(S \Rightarrow T)} = \left(\mu_S(x) \Rightarrow \mu_T(x)\right)$$
$$\mu_{(S \setminus T)}(x) = \max\left(0, \mu_S(x) - \mu_T(x)\right)$$
$$S \subseteq T \ \text{ iff } \mu_S(x) \le \mu_T(x) \ \text{ for all } x \in U$$

Подписано в печать 10.11.2016 г.

| | | |
|---|---|---|
| Печать офсетная | Формат 60х84/16 | 0,41 уч.-изд. л. |
| 0,7 усл.печ.л. | Заказ № 118 | Тираж 60 экз. |

Московский государственный технический университет ГА
*125993 Москва, Кронштадтский бульвар, д. 20*
Редакционно-издательский отдел