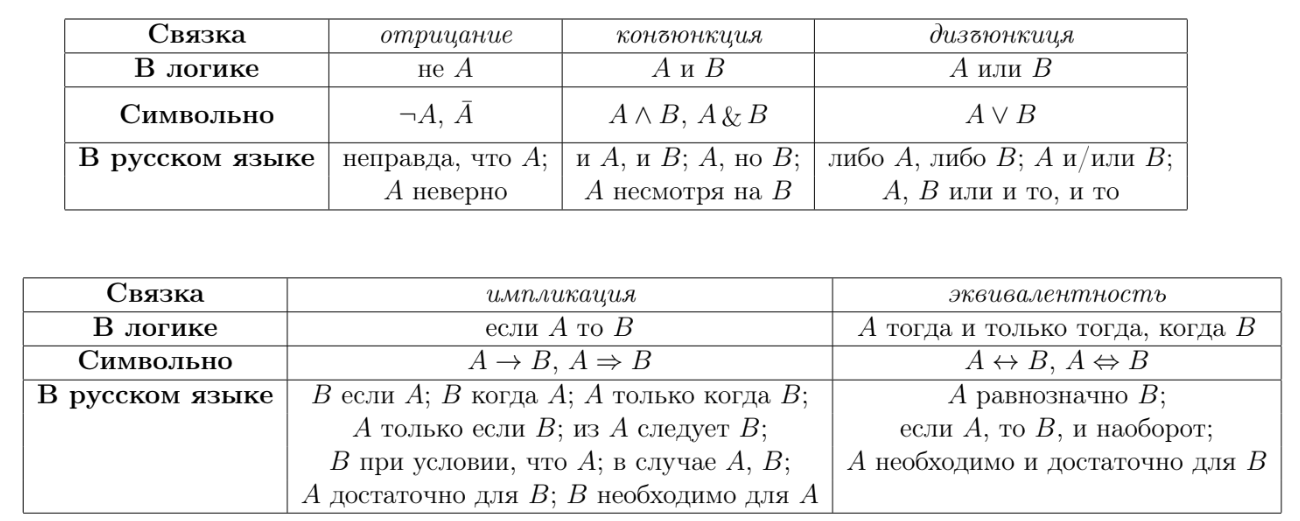
**1. Истинность, ложность и высказывания**





**D/Д** – предметная область предикатов

Конъюнкция и дизъюнкция имеют меньший ***приоритет***, чем отрицание, но больший, чем импликация.

Высказывания F и G (логически) **эквивалентны** тогда (и только тогда), когда они оба истинны либо оба ложны, в зависимости от истинностных значений высказываний A1, . . . , An.

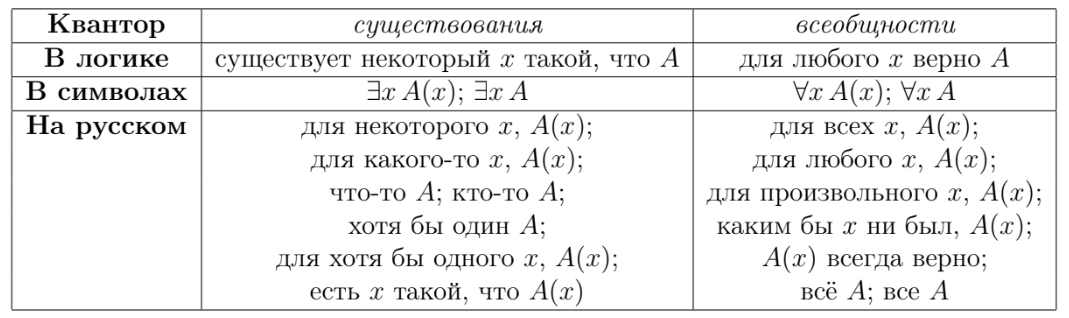
Мы будем обозначать данное отношение как F ≡ G.

Высказывание F, состоящее из высказываний A1, . . . , An мы называем **тавтологией** если оно (всегда) истинно при любых значениях высказываний A1,. . . , An.

*Чтобы доказать тавтологичность формулы, достаточно предположить, что она ложна и прийти к противоречию.*

Рассуждение **корректно** тогда и только тогда, когда вывод истинен при истинности посылок.

**2. Язык математики**

****

¬ ¬A≡ A A∨B≡ B∨A (аналог. с ∧)

¬(A∧B)≡ ¬A ∧¬B ¬(A∨ B)≡ ¬A ∨ ¬B

**A→B≡¬A∨B** **¬ (A→B)≡ A∧¬B** A→B≡¬B→¬A

**A→(B →С) ≡ (A∧B) →С**

∃x A(x) ≡ ¬∀x ¬A(x) ∀x A(x) ≡ ¬∃x ¬A(x)

¬∀x (Р1(x) → P2(x)) ≡∃x (Р1(x) ∧ ¬P2(x))

Связывание квантора сильнее (имеет больший приоритет), чем любая логическая связка, так что область действия квантора заканчивается на первой бинарной логической связке справа от него. Чтобы включить связку в область действия квантора, необходимы скобки: например, в ∀x (A(x) → B(x)). С другой стороны, в формуле ∀x A(x) → B(x), никакое вхождение x в B(x) не связано первым квантором. То есть любое такое вхождение свободно для подстановки вместо него значения (если не связано каким-либо квантором внутри B(x)).

Vacuous truth (пустая истина). Пример: Сегодняшний король Франции лысый. ∀x(K(x) → B(x)) – всегда истина, так как посылка всегда ложна.

Стандартный алгоритм доказательства утверждения «не существует такого x, что A(x)»: рассмотрим произвольный x такой, что A(x) истинно, и докажем, что какое-то известно ложное высказывание логически следует из A(x). Следовательно, если бы x с истинным A(x) существовал, то заведомо ложное высказывание было бы истинным, что невозможно. Таким образом, такого x не существует.

**3. Cтроки/Списки**

Правила образования списков под алфавитом А:

(П1) [ ] — строка над A; **[ ] ∈ S(A);**

(П2) если s′ — строка над A, а x — символ из A, тогда x : s′ — тоже строка над A, **∀s′ ∀x ( (s′ ∈ S(A) ∧ x ∈ A) → x : s′ ∈ S(A) )**

(А1) **[ ] ≠ x : s** для всех x и s.

(А2) **x : s = y : t ⇐⇒ (x = y ∧ s = t)** для всех x, y ∈ A и s, t ∈ S(A), т.е. две строки, составленные с помощью оператора конкатенации **:** равны тогда и только тогда, когда соответствующие их головы и хвосты равны.

**Принцип индукции** (ПИ) для строк: Для любого унарного предиката P над S(A), если P([ ]) и ∀s∀x (P(s) → P(x : s)), то ∀s P(s).

Утверждение P([ ]) называется базой индукции, ∀s∀x (P(s) → P(x : s)) — индукционным переходом, а P(s) — предположением индукции.

**Рекурсивные функции для строк:**

**ln(s) –** *длина строки***.** Определим ln следующими двумя правилами:

ln([ ]) = 0; ln(x : s) = 1 + ln(s) для всех x и s.

**app(s1, s2)** – *конкатенация*/*склеивание*. Определение:

app([ ], t) = t для любого t; app(x : s, t) = x : app(s, t) для любых x, s, и t.

**rev(s)** – *обращение/инверсия строки*. Определение:

rev([ ]) = [ ]; rev(x : s) = app(rev(s), [x]) для всех x и s.

**init(s)** – *удаление последнего символа строки*. Определение:  
init([ ]) = [ ]; init(x:[ ]) = [ ] для любого x из алфавита А

init(s≠[ ] → init (x:s) = x:init(s)) для любого символа x из алфавита А и любой строки s, состоящей из символов алфавита А

**Леммы для строк:**

1) Для любых s ∈ S(A), app(s, [ ]) = s.

2) Для любых s, t, r ∈ S(A), app(s, app(t, r)) = app(app(s, t), r).

3) Для любых s, t, r ∈ S(A), если app(s, t) = app(s, r), то t = r.

4) Для любых s, t ∈ S(A), если app(s, t) = [ ], то s = [ ] и t = [ ].

**4. Индукция:** ∀φ (φ в определениях фиксирован)

**ПМИ:** **φ(0) ∧ ∀n ∈ N (φ(n) → φ(n + 1))→ ∀n ∈ N φ(n)**. (φ фиксирован)

Для произвольного предиката φ, если выполнено φ(0) и для каждого n ∈ N φ(n) влечёт φ(n + 1), то φ(n) выполнено для всех n ∈ N.

Утверждение φ(0) называется *базой индукции*, ∀n ∈ N (φ(n) → φ(n + 1)) - *шаг индукции*, а утверждение φ(n) для каждого n - *предположение индукции*.

**ПСИ: ∀n (∀m<n φ(m) → φ(n)) → ∀n φ(n)** или **Prog(φ)→ ∀n φ(n)**

Для произвольного предиката φ, если верно φ(0) и для каждого n ∈ N φ(n + 1) выполнено, когда выполнены φ(0), φ(1). . . , φ(n), то φ(n) верно для любого n ∈ N.

Он позволяет нам использовать в качестве индукционного предположения не одно лишь φ(n), а все утверждения φ(0), φ(1), . . . , φ(n) вместе.

**ПНЧ:** **∃n φ(n) → ∃n (φ(n) ∧ ∀m<n ¬φ(m)**

Если существует хотя бы одно натуральное число (n), для которого выполняется (φ(n)), то существует наименьшее такое число (n), для которого выполняется (φ(n)), и для всех чисел, меньших (n), утверждение (φ(m)) не выполняется.

**5. Делимость и деление** (подразумеваем ***целые*** числа)

***Делимость:*** число a делит число b, если существует k ∈ Z такое, что **b = ak**. В свою очередь, число b делится на число (или числом) a. Также мы называем a делителем числа b, а b — кратным числу a. Мы будем писать **a | b**, если a делит b. Пример: 1; −2 | 6 т.к. 6 = (−2) · (−3);

**Леммы для делимости:**1) ±1 | a (a=±1\*a) 2) 0 | a (0=a\*0)

3) a|1 => a=±1 a|-1 => a=±1 4) a | a;

5) если a | b и b | c, то a | c; 6) если a | b и b | a, то a = ±b.

7) если a | b и a | c, то a |(b + c) и a |(b − c)

Следствие Л7: если a |(b + c) или a |(b − c) и a | b, то a | c.

***Деление:*** Для любых натуральных чисел a и b **≠** 0, существует уникальная пара (q, r) ∈ N^2 такая, что a = bq + r и 0 ≤ r < b.  
Такое число q называется *неполным частным*, а r — *остатком от деления* a на b.

Следствие. Для всех чисел a и b ̸= 0, существует уникальная пара   
(q, r) ∈ Z × N такая, что a = bq + r и 0 ≤ r < |b|.

***Модульная арифметика***. Положим m - положительное число. Тогда   
a сравнимо с b по модулю m, если m |(a − b). Мы записываем это как   
**a ≡ b (mod m)** или **a ≡ b (m)** и называем m ***модулем***.   
m > 1 (т.к. любое число делится на 1)  
*Лемма*: Для любых a, b, m, верно, что a ≡ b (mod m) тогда и только тогда, когда a и b дают одинаковый остаток при делении на m.

Следствия:   
1) Никакие два числа из 0, 1, . . . , m − 1 не сравнимы по модулю m.   
2) Предположим r — остаток при делении a на m. Тогда a ≡ r (mod m).   
3) Для любых a, b, m верно следующее:

* a ≡ a (mod m);
* если a ≡ b (mod m), то b ≡ a (mod m);
* если a ≡ b (mod m) и b ≡ c (mod m), то a ≡ c (mod m).

Положим a ≡ b (mod m) и c ≡ d (mod m). Тогда верно следующее:

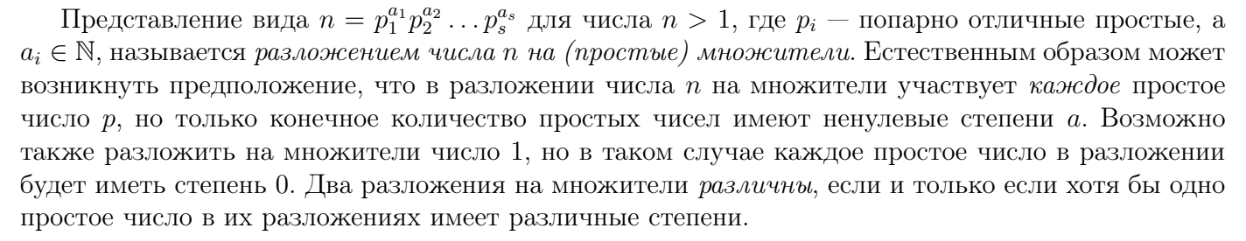
1. a + c ≡ b + d (mod m);

2. ac ≡ bd (mod m);

3. a^n ≡ b^n(mod m) для всех n ∈ N.

***Простые числа и разложение на множители***. Число p > 1называется **простым**, если оно делится на ±1, на ±p и ни на что больше. В противном случае, число больше 1 зовется **составным**.

Отрицательные числа, 0 и 1 не являются ни простыми, ни составными.



***Основная теорема арифметики***: Для любого числа n > 1 существует уникальное его разложение на простые множители.

*Теорема.* Существует бесконечно много простых чисел.

***Признак делимости.*** 