**Математическая логика**

**1. Высказывания: простые, сложные, примеры высказываний. Истинностное значение высказывания.**

Высказывание – утверждение или повествовательное предложение, о котором можно однозначно сказать истинно или ложно.

Примеры высказываний:

2\*2=4 – истина

1 – отрицательное число – ложь

Простые высказывания – высказывания, которые не содержат логических связок.

Сложные высказывания – высказывания, которые содержат логические связки.

Истинностное значение высказывания – истинность или ложность, приписываемые высказыванию.

**2. Определение логичного рассуждения. Примеры.**

Рассуждение называется логичным, если всякий раз, когда все посылки истинны, заключение также истинно, то есть из истинности конъюнкции всех гипотез следует истинность заключения.

Примеры:

**3. Логичные рассуждения и проверка на тавтологию (как связаны).**

Рассуждение P1, P2,…, Pn |- D логично (а клауза – верная) тогда и только тогда, когда формула P1 Ʌ P2 Ʌ …Ʌ Pn -> D является тавтологией.

**4. Логичные рассуждения и проверка на противоречие (как связаны).**

Рассуждение P1, P2,…, Pn |- D логично (а клауза – верная) тогда и только тогда, когда формула P1 Ʌ P2 Ʌ …Ʌ Pn Ʌ (not D) является противоречием.

**5. Дать определение конъюнкции и импликации, дизъюнкции и эквиваленции.**

Конъюнкция – Пусть A, B – некоторые высказывания. Тогда конъюнкция A&B – сложное высказывание, которое истинно ТИТТ, когда А, В – одновременно истинны.

Импликация – Пусть A, B – некоторые высказывания. Тогда импликация A->B – сложное высказывание, которое ложно ТИТТ, когда А - истинно, а В – ложно.

Дизъюнкция – Пусть A, B – некоторые высказывания. Тогда дизъюнкция Aor B – сложное высказывание, которое ложно ТИТТ, когда А, В – одновременно ложны.

Эквиваленция – Пусть A, B – некоторые высказывания. Тогда эквиваленция A=B – сложное высказывание, которое истинно ТИТТ, когда истинностные значения А, В – одинаковы.

**6. Определение формулы алгебры высказываний.**

Алфавит А:

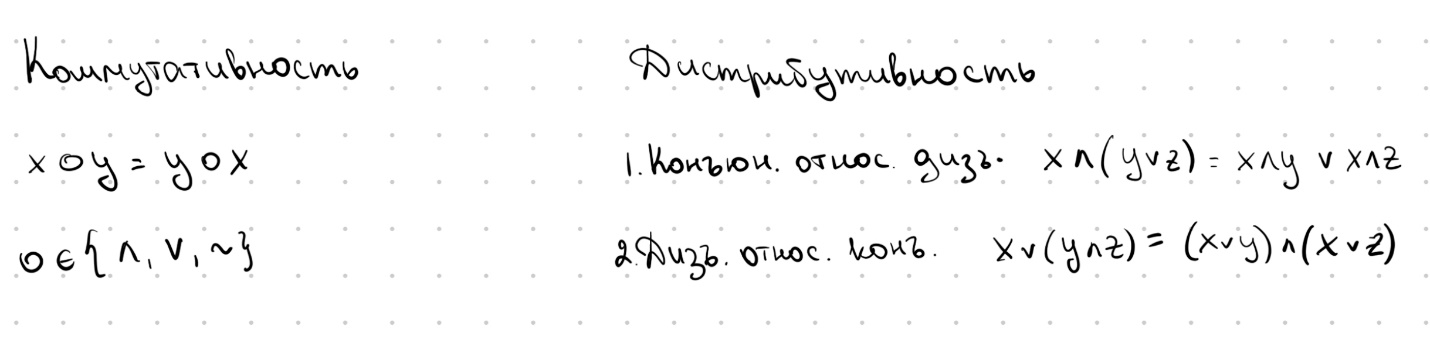
-символы высказывательных переменных x, y, z.

-символы логических связок not, V, Ʌ, ~, ->

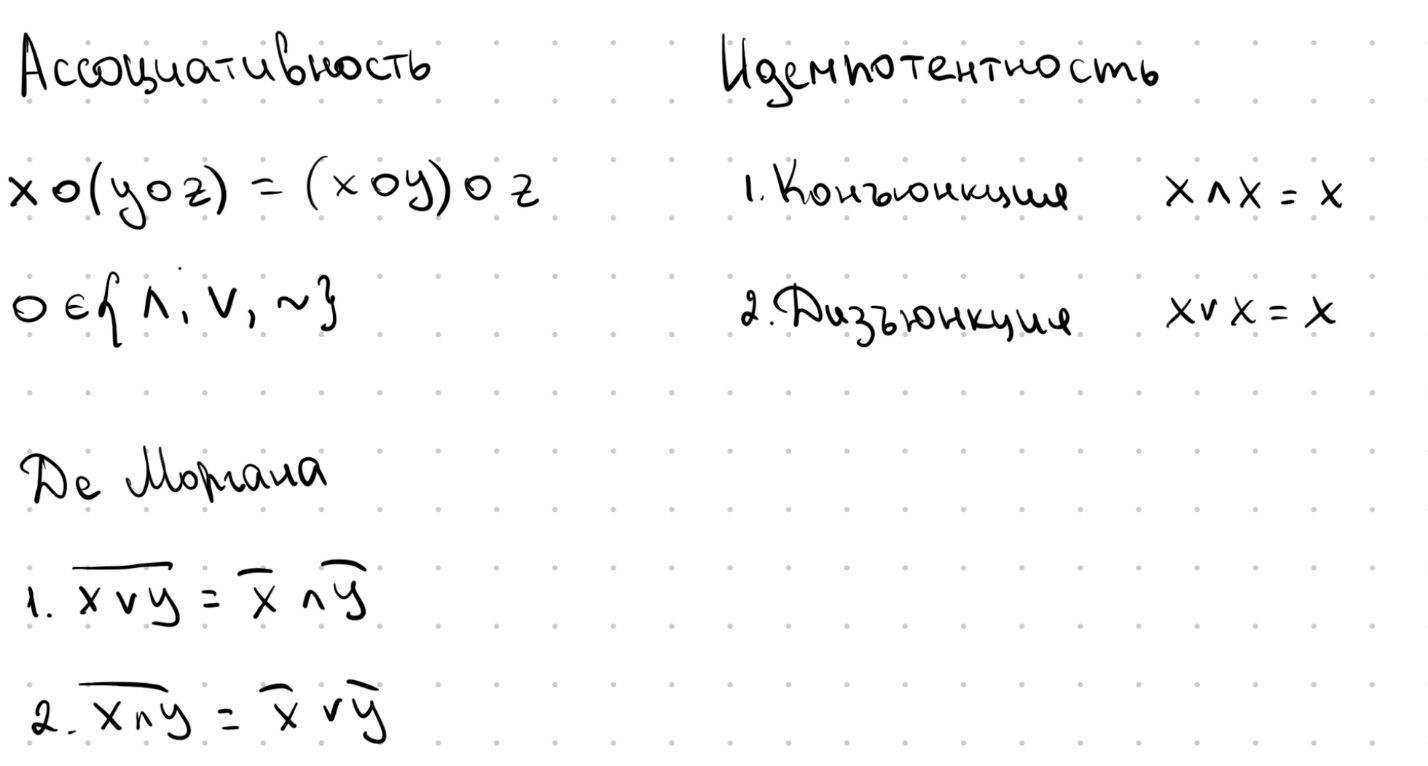
-скобки()

Слово – любая конечная последовательность символов алфавита.

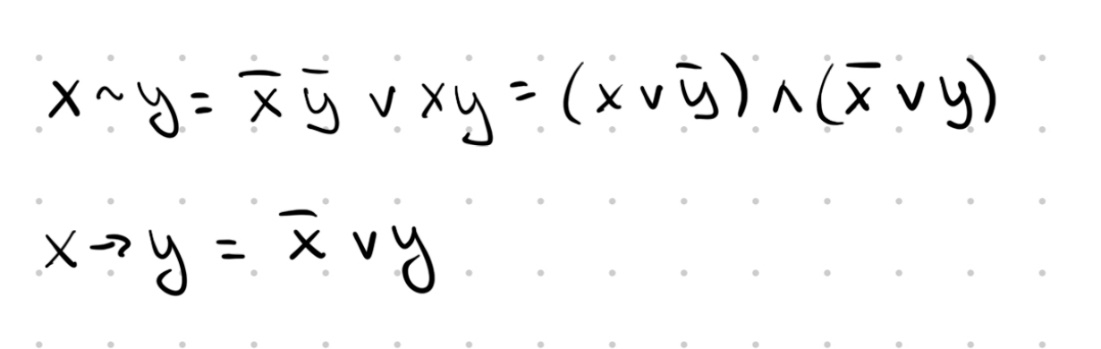
**7. Свойство коммутативности и дистрибутивности. Какие логические связки им удовлетворяют?**



**8. Свойство ассоциативности и идемпотентности. Какие логические связки им удовлетворяют? Правила де Моргана.**

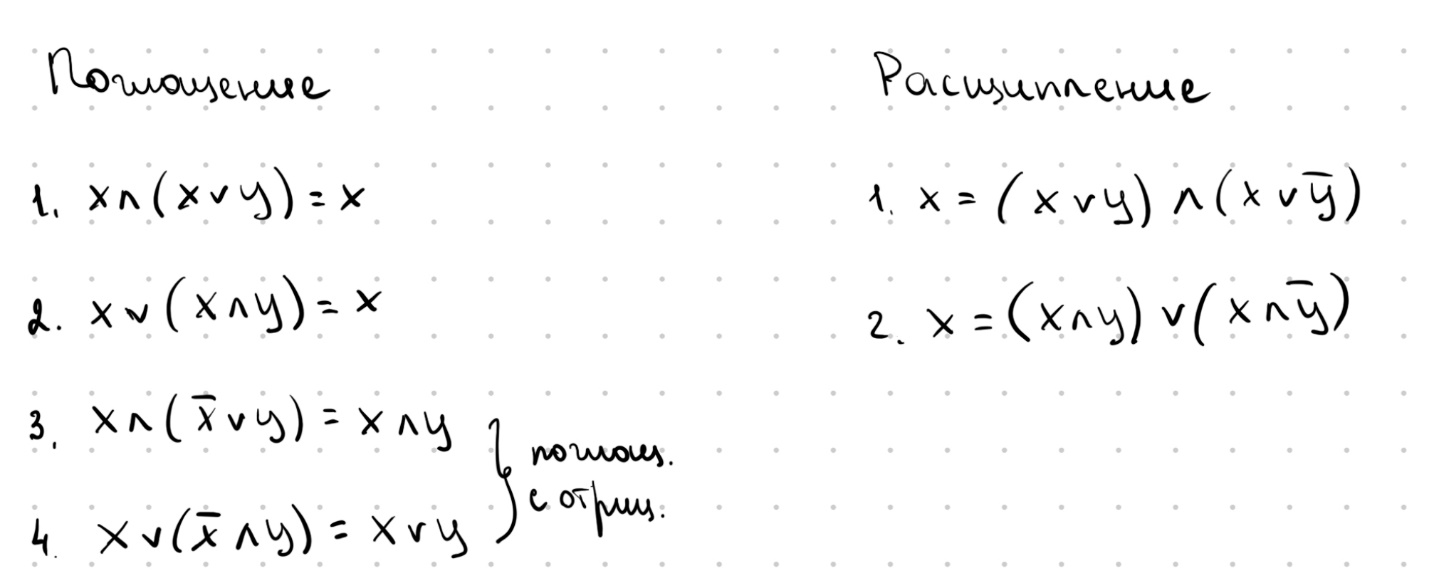


**9. Представление эквиваленции и импликации булевыми формулами.**



**10. Равносильные формулы. Формулы поглощения и расщепления.**

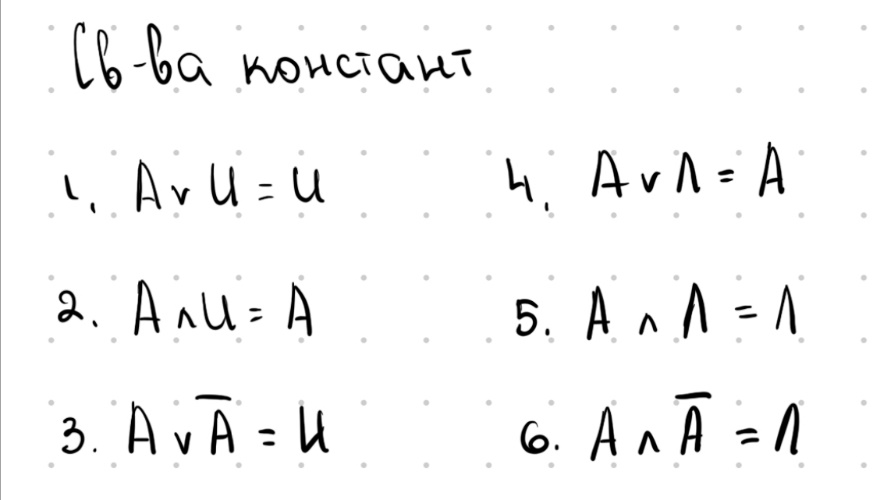
Две формулы Ф1, Ф2 логики высказываний, зависящие от одних и тех же переменных, равносильны (Ф1 = Ф2), если таблицы задаваемых ими функций совпадают.



**11. Тавтология и противоречие. Свойства констант (6 законов).**

Формула называется тавтологией, или тождественно истинной, если задаваемая ею функция истинна при всех значениях высказывательных переменных. Ф=И – тавтология.

Формула называется противоречием, или тождественно ложной, если задаваемая ею функция ложна при всех значениях высказывательных переменных. Ф=Л – противоречие.



**12. Двойственная формула. Принцип двойственности.**

Формула Ф\*называется двойственной по отношению к булевой формуле Ф, если:

-все вхождения Ʌ в формуле Ф заменены на V в Ф\*

-все вхождения V в Ф заменены на Ʌ в Ф\*.

Принцип двойственности: две формулы равносильны, то есть Ф1 = Ф2 тогда и только тогда, когда равносильны двойственные к ним, то есть Ф1\* = Ф2\*.

**13. Проблема разрешимости в логике высказываний.**

Существует ли процедура, которая за конечное число шагов позволяет выяснить, является ли произвольная формула логики высказываний тавтологией, или нет?

Проблема разрешимости алгоритмически разрешима:

1.Способ проверить на тавтологию: построить таблицу истинности формул. Недостаток: размерность таблиц растет с ростом числа.

2.Способ проверить на тавтологию: построить ДНФ формулы и проверить на противоречие.

3.Способ проверить на тавтологию: построить КНФ формулы.

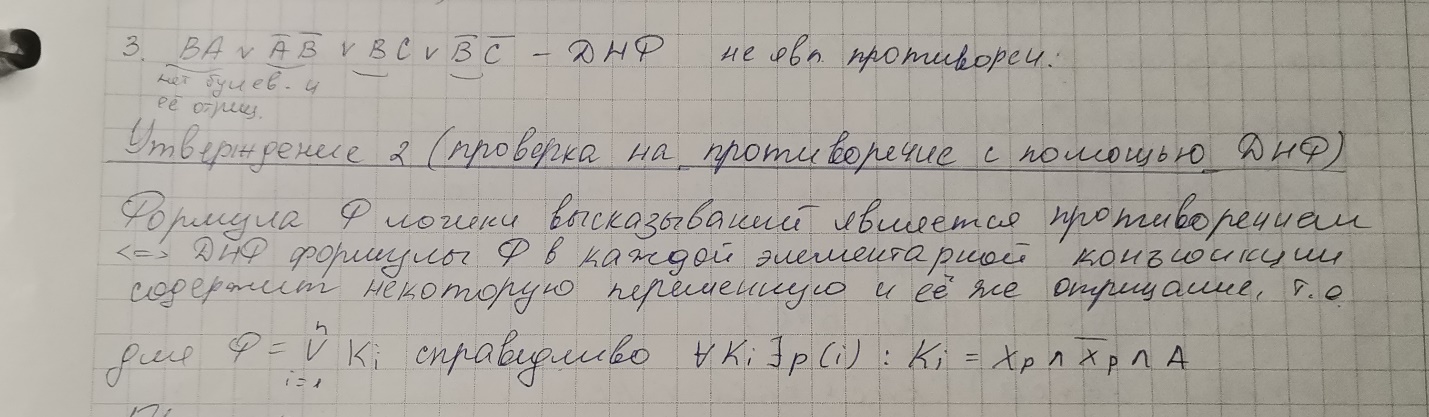
**14. Определение КНФ. Как проверить, что формула является тавтологией с помощью КНФ?**

Конъюнктивно нормальная форма – конъюнкция конечного числа элементарных дизъюнкций.

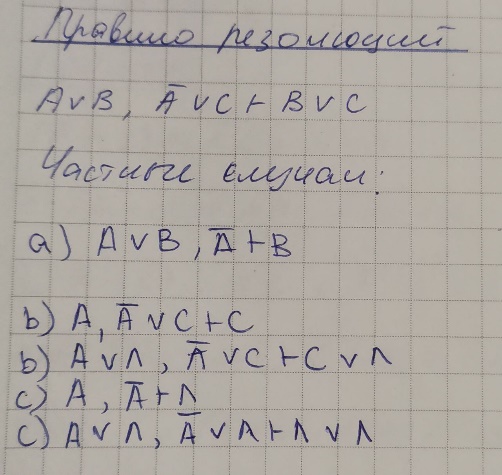
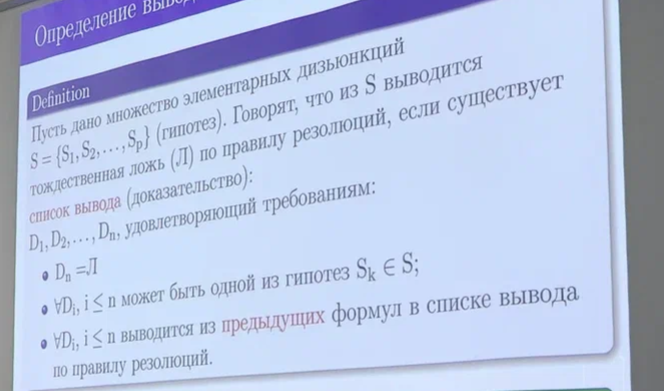


**15. Определение ДНФ. Как проверить, что формула является противоречием с помощью ДНФ?**

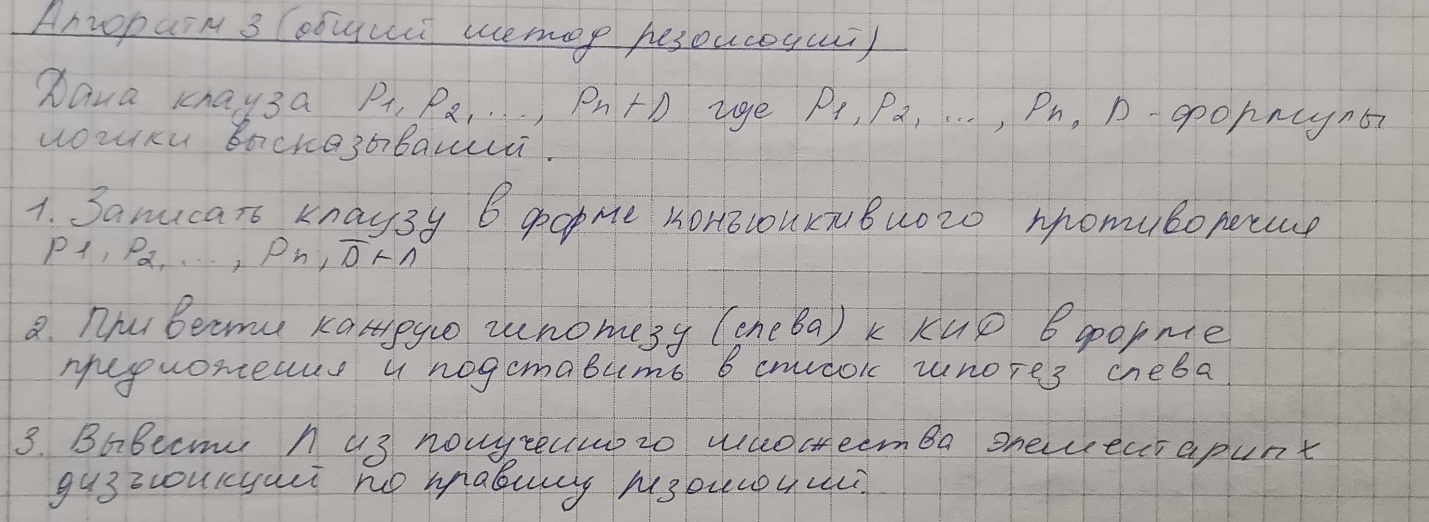
Дизъюнктивно нормальная форма – дизъюнкция конечного числа элементарных конъюнкций.



**16. Правило резолюций. Вывод формулы по правилу резолюций в логике высказываний (определение).**

**17. Алгоритм метода резолюций в логике высказываний.**



**18. Как задать аксиоматическую теорию (4 пункта).**

Аксиоматическая теория задана, если заданы

-задан алфавит А символов

-определено понятие формулы над алфавитом А

-из множества всех формул выделена группа формул – аксиом ФАТ

-заданы правила вывода формулы в данной теории

**19.Определение вывода формулы в аксиоматической теории.**

Пусть дано множество формул P = {P1, P2, …, Pm} (гипотез). Говорят, что из Р выводится формула D по правилам вывода АТ, если существует список вывода (доказательство):

D1, D2,…, Dn, удовлетворяющий требованиям:

- Dn = D

- ⱯDi, i <= n может быть одной из гипотез Pk ∈ P

- ⱯDi, i <= n может быть одной из аксиом (теорем) теории

- ⱯDi, i <= n выводится из предыдущих формул в списке вывода по правилам вывода данной теории

В этом случае клауза P1, P2,…,Pn |- D считается доказанной в АТ.

**20.Правила подстановки и m.p.**

Правило подстановки - Если формула Ф – аксиома (теорема) теории, а А – переменная данной формулы, то после замены всех вхождений переменной А на произвольную формулу Ф1, получим теорию данной теории.

Правило m.p. – А, А -> B |- B

Работа правила в списке вывода: D1, D2,…, A, Dk, Dk+1,…, A->B,…,B,…

**21.Теорема дедукции.**

Если Р – список гипотез и известно, что P, A |- B, то доказано, что P |- A -> B.

**22. Правила силлогизма и разъединения посылок.**

Правило силлогизма

-Правило силлогизма для формул: A -> B, B -> C |- A -> C

-Правило силлогизма для теорем: Если доказаны теоремы |- A -> B, |- B -> C, то доказана теорема |- A -> C

-По теореме дедукции и правилу m.p. правило силлогизма ⬄ теореме 3 |- (A->B)->((B->C)->(A->C)).

Правило разъединения посылок

A Ʌ B -> C |- A -> (B -> C)

**23. Правила перестановки и соединения посылок.**

Правило перестановки

-Правило перестановки для формул: A -> (B->C) |- B -> (A -> C)

-Правило перестановки для теорем: Если доказана теорема |- A->(B->C), то доказана теорема |- B->(A->C).

Правило соединения посылок

A -> (B -> C) |- A Ʌ B -> C

**24. Полнота и непротиворечивость аксиоматической теории (определения). Независимость системы аксиом.**

Аксиоматическая теория является полной, если любая формула, выражающая логический закон, является теоремой данной теории.

Аксиоматическая теория является непротиворечивой, если не существует такой формулы A, что A и ¬A одновременно теоремы.

Система аксиом называется независимой, если не существует аксиомы, выводимой из остальных.

**25. Определение предиката. Предметное множество и предметные переменные. Местность предиката.**

Предикат – Функция P(x1, x2,…, xn) от n переменных называется предикатом, если Ɐ (x1, x2,…, xn) P(x1, x2,…, xn) ∈ {И, Л}.

Переменные (x1, x2,…,xn) предиката P(x1, x2,…, xn) называются предметными переменными, каждая из них принимает свои значения из некоторого предметного множества Mi, i = 1, 2, 3,…

**26. Связывание кванторами всеобщности и существования (определение, примеры).**

Квантор всеобщности:

Пусть P(x) – одноместный предикат, где x ∈ M. Тогда, по определению, (Ɐx)P(x) - нуль-местный предикат, который принимает значение И, если P(x) = И для каждого x ∈ M.

Пример: P(x) “x делится на 3” М = {3, 7, 9} (Ɐx)P(x) = Ложь, тк есть 7: P(7) = Л.

Квантор существования:

Пусть P(x) – одноместный предикат, где x ∈ M. Тогда, по определению, (∃x)P(x) - нуль-местный предикат, который принимает значение И, если существует такой элемент а ∈ М такой, что Р(а) = И.

**27. Интерпретация формулы. Равносильность формул в данной интерпретации.**

Интерпретация формулы задана, если:

-Для каждой предметной переменой заданы конкретные предметные множества.

-Задано соответствие f, которое каждому предикатному символу данной формулы сопоставляет конкретный предикат.

Равносильность формул в данной интерпретации:

Две формулы Ф1, Ф2 логики предикатов, зависящие от свободных переменных х1, х2,…,хк, равносильны в заданной интерпретации, если на любом наборе значений свободных переменных х1, х2,…,хn их значения одинаковы, то есть Ф1(х1, х2,…,хк) = Ф2(х1, х2,…,хк)

**28. Равносильность формул на множестве и в логике предикатов. Перенос квантора через отрицание.**

Равносильность формул на множестве:

Две формулы Ф1, Ф2 логики предикатов равносильны на конкретных предметных множествах М1, М2,…,Мn их предметных переменных, если они равносильны на любых интерпретациях, заданных на М1, М2,…,Мn.

Равносильность формул в логике предикатов:

Две формулы Ф1, Ф2 логики предикатов равносильны в логике предикатов, если они равносильны на любых предметных множествах М1, М2,…,Мn их предметных переменных.

Перенос квантора через отрицание:

Пусть х – свободная переменная формулы Р(х) (остальные переменные не указаны), тогда

- ¬ ((Ɐx)P(x)) = (∃x) ¬P(x)

- ¬ ((∃x)P(x)) = (Ɐx) ¬P(x)

**29. Правила вынесения квантора за скобки.**

Пусть х – свободная переменная формулы Р(х), Q(x) (остальные переменные не указаны), тогда

- (Ɐx)P(x) Ʌ (Ɐx)Q(x) = (Ɐx)(P(x) Ʌ Q(x))

- (∃x)P(x) V (∃x)Q(x) = (∃x)(P(x) V Q(x))

**30. Приведенная и нормальная формы формул (определение).**

Приведенная форма формул.

Формула находится в приведенной форме, если она булева и знаки отрицания стоят только над символами предикатов.

Нормальная форма формул.

Приведённая форма, либо не содержащая кванторов, либо все кванторы впереди формулы.

**31. Выполнимость и общезначимость формулы логики предикатов.**

Выполнимость формулы логики предикатов.

Формула А логики предикатов называется выполнимой в области М, если существуют значения переменных входящих в эту формулу и отнесенных к области М (иначе – существует модель), при которых формула А принимает истинные значения.

Общезначимость формулы логики предикатов.

Формула А логики предикатов называется общезначимой, если она тождественна истинна на всякой области (на любой модели).

Если две равносильные формулы логики предикатов соединить знаком эквиваленции https://studfile.net/html/2706/20/html_TTBtwW1mOv.na2b/htmlconvd-UYxM1T_html_77217b73c8eab5f5.gif , то полученная формула будет принимать значение И для любого набора переменных в любой области, т.е. будет общезначимой.

**32. Проблема разрешимости в логике предикатов. Теорема Черча-Тьюринга.**

Проблема разрешимости в логике предикатов.

Проблема разрешимости в логике предикатов состоит в нахождении алгоритмической процедуры, с помощью которой для каждого правильно построенного предложения можно решить, выводимо оно или нет, т.е. можно ли его доказать в логике предикатов.

В логике предикатов проблема разрешимости в общем случае не имеет решения, т.е. не существует алгоритмической процедуры, позволяющей определить, выводима ли данная формула или нет.

Теорема Чёрча-Тьюринга.

Не существует алгоритма, который для произвольной формулы логики предикатов устанавливал бы, общезначима она, или нет.

**Теория алгоритмов**

**1. Элементарные рекурсивные функции.**



**2. Оператор Snm(…)**

Пусть φ(t1, t2,…., tm), fi(x1, x2,….,xn), i = 1, 2,…,m – частичные числовые функции, тогда частичная числовая функция f(x1, x2,….,xn) называется результатом применения оператора суперпозиции Snm(φ, f1, f2,…,fm) к функциям φ(t1, t2,…., tm), fi(x1, x2,….,xn) ), i = 1, 2,…,m, если f(x1, x2,….,xn) = Snm(φ, f1, f2,…,fm) = φ(f1(x1, x2,….,xn), f2(x1, x2,….,xn),…,fm(x1, x2,….,xn)), т.е. φ(f1, f2,…., fm) = f.

**3. Общая схема примитивной рекурсии для функции 1 переменной.**

F(x) – результат применения оператора примитивной рекурсии к частично числовым функциям φ и ψ(x, y) если она удовлетворяет следующей схеме примитивной рекурсии

Пр: { f(0) = φ

f(x +1) = ψ(x, f(x))

**4. Общая схема примитивной рекурсии для функции 2 переменных.**

f(x, y) – результат применения оператора примитивной рекурсии к частично числовым функциям φ(x) и ψ(x, y, z) если она удовлетворяет следующей схеме примитивной рекурсии (по переменной у).

Пр: { f(x, 0) = φ(x)

f(x, y+1) = ψ(x, y, f(x, y))

**5. Какая функция называется примитивно-рекурсивной.**

Частичная числовая функция f называется примитивно-рекурсивной, если она может быть получена из трех элементарных рекурсивных функций (O(x), S(x), Inm(x1, …, xn)) с помощью операторов суперпозиции и примитивной рекурсии, примененных конечное число раз. Обозначение: f ∈ Kпр

**6. Оператор минимизации для функции 1 переменной.**

Функция g(x) является результатом применения оператора минимизаци (по переменной х) к частично числовой функции f(x), т.е. g(x) = mxf(x), если для каждого фиксированного x = a ∈ N функция g(x) строится следующим образом:

1.Составляем уравнение минимизации: (\*) f(y) = a.

2.Находим минимальный корень уравнения (\*) y0 ∈ N (если такой существует)

3.Если для любого y < y0, y ∈N функция f(y) определена, то g(a) = y0, т.е. y0 – значение g(x) при x = a.

4.Если хотя бы одно из условий п. 2-3 не выполнено, то g(x) не определена при х = а.

**7. Какая функция называется частично-рекурсивной.**

Частичная числовая функция f называется частично-рекурсивной, если она может быть получена из трех элементарных рекурсивных функций (O(x), S(x), Inm(x1, …, xn)) с помощью операторов суперпозиции, примитивной рекурсии и минимизации, примененных конечное число раз.

**8. Функции информационной ленты машины Тьюринга.**

-Лента разбита на ячейки. В любой момент времени t в каждой ячейке может быть записан ровно один символ из внешнего алфавита машины Т A = {a0, a1,…,an}.

-Во внешнем алфавите А выделена особая роль пустого символа a0. Ячейка называется пустой, если в ней записан пустой символ. Лента, состоящая из пустых ячеек, называется пустой.

-Основная функция информационной ленты – хранение информации.

**9. Функции считывающей-записывающей головки машины Тьюринга.**

-В любой момент времени t головка может обозревать ровно одну ячейку на ленте.

-Основными функциями головки являются чтение символа из внешнего алфавита, содержащегося в ячейке, и запись символа из внешнего алфавита в ячейку.

**10. Конфигурация на ленте машины Тьюринга.**

Конфигурация на ленте в момент времени t задана, если:

-Задано машинное слово Р, записанное в ячейках ленты

-Задано положение головки на ленте (обозреваемый символ)

-Задано внутреннее состояние из множества Q

**11. Команда машины Тьюринга.**

Пятерка символов qi, aj, ql, ak K называется командой машины Т, где qi ∈Q – внутреннее состояние управляющего устройства машины в момент времени t.

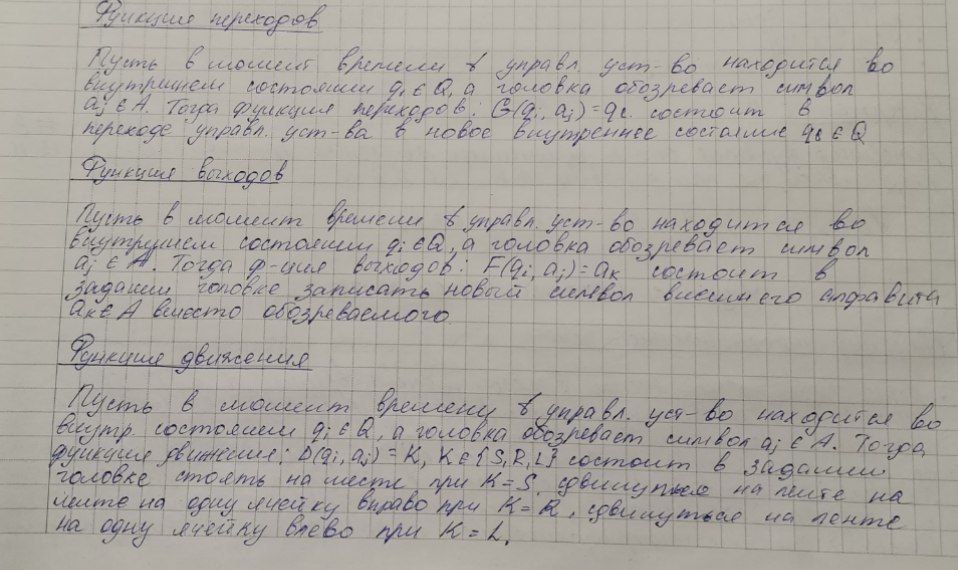
aj ∈А – обозреваемый головкой символ в момент времени t.

ql ∈А – новое внутренне состояние, в которое нужно перейти по команде.

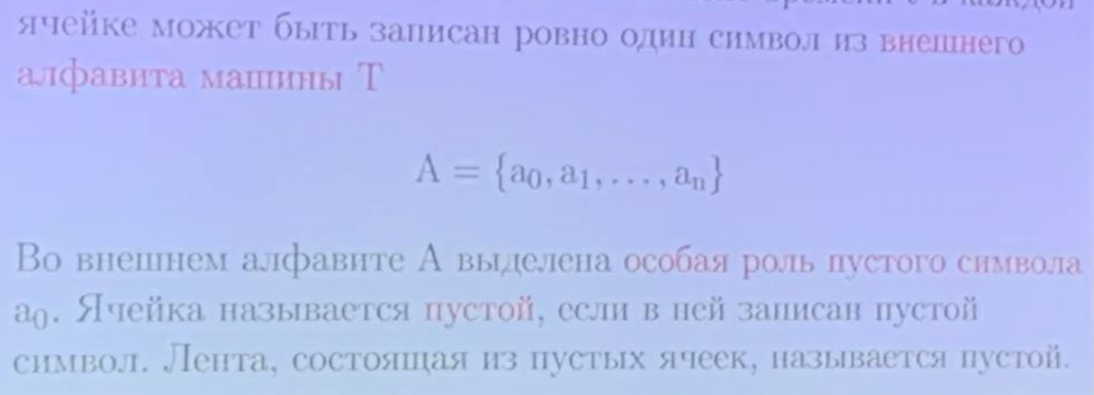
ak ∈А – новый символ, который нужно записать в место обозреваемого.

K – движение, которое нужно совершить по данной команде (K = S, R, L).

**12. Функции управляющего устройства машины Тьюринга.**



**13. Внешний алфавит и внутренние состояния машины Тьюринга.**





**14. Программа машины Тьюринга.**

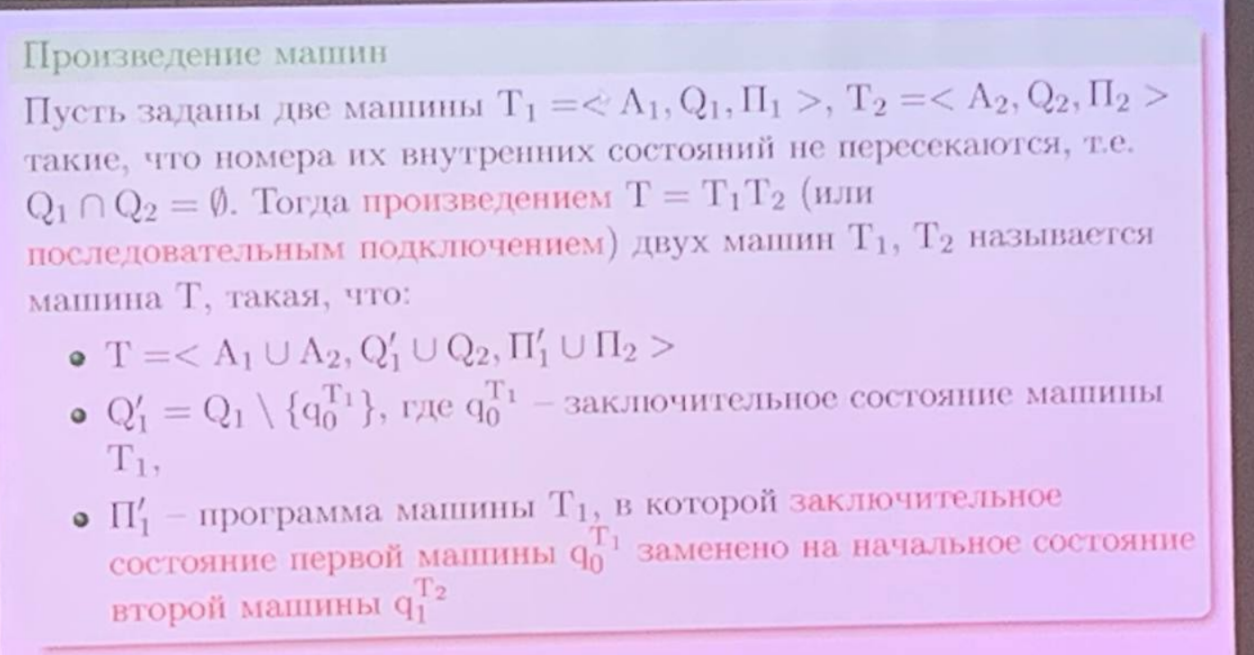
Программа машины Т (П) – совокупность команд, которые может выполнить данная машина Т. Любая машина задается триплетом Т = <A, Q, П>, где А – внешний алфавит, Q – внутренний алфавит, П – программа машины Т.

**15. Применимость и неприменимость машины Тьюринга к данному слову.**

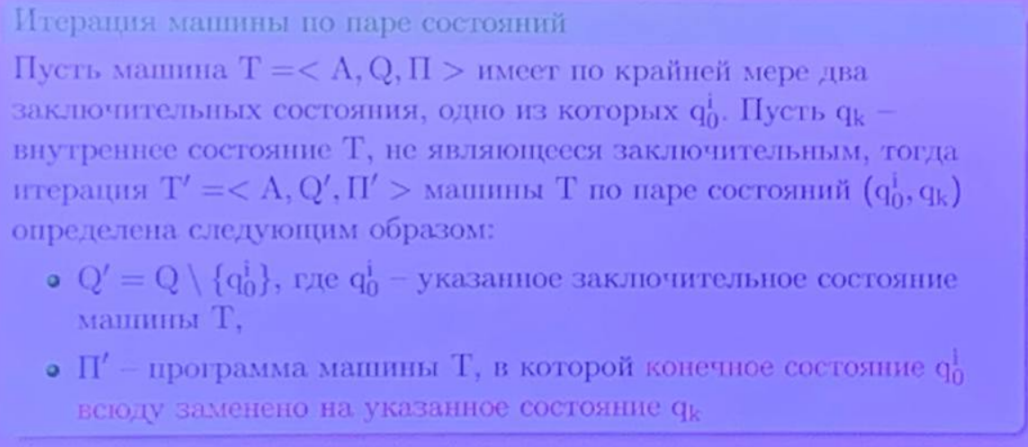
*Определение применимой к слову машины Т* – Машина Т называется применимой к входному слову Pl, если, начав работу в начальной конфигурации с данным словом, она остановится в заключительной конфигурации с выходным словом P0. Тогда P0 = T(P1).

*Неприменимая к лову машина Т* – Машина Т называется неприменимой к входному слову Pl, если начав работу в начальной конфигурации с данным словом, она никогда не остановится (зациклится).

**16. Произведение машин Тьюринга.**



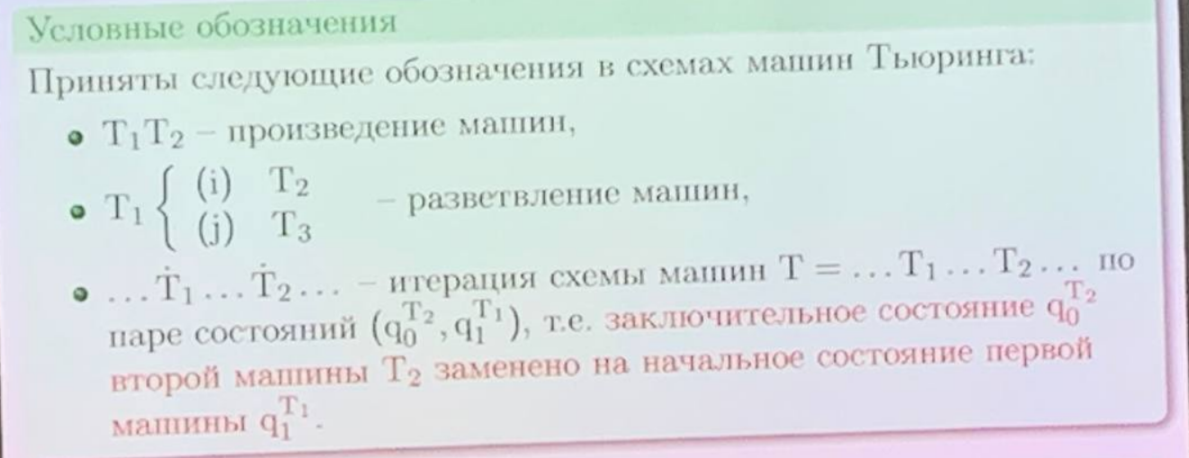
**17. Итерация машины Тьюринга по паре состояний.**



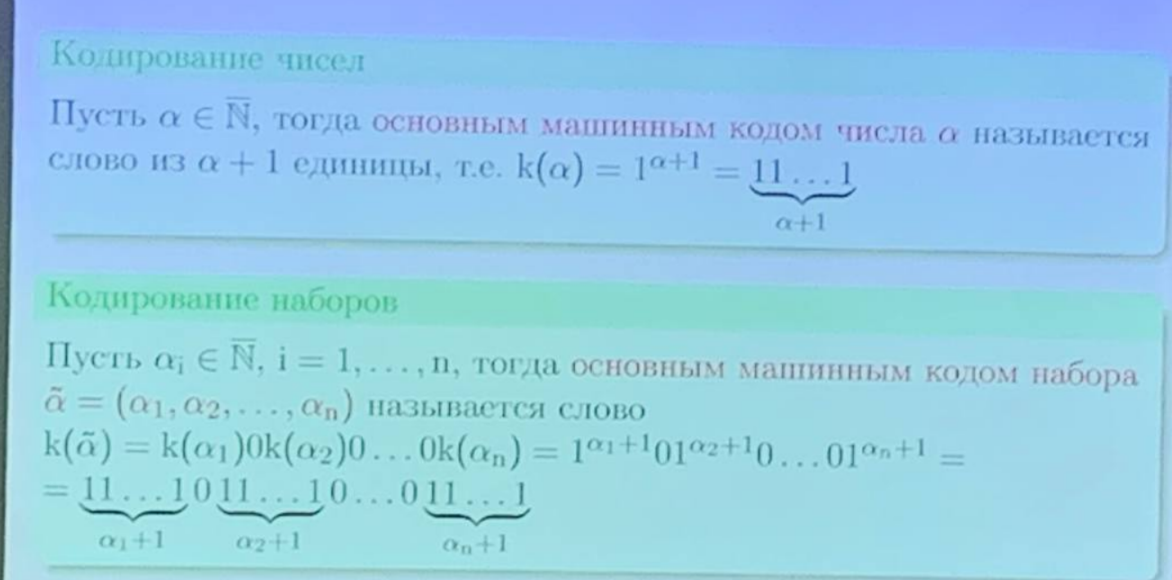
**18. Разветвление машин Тьюринга.**



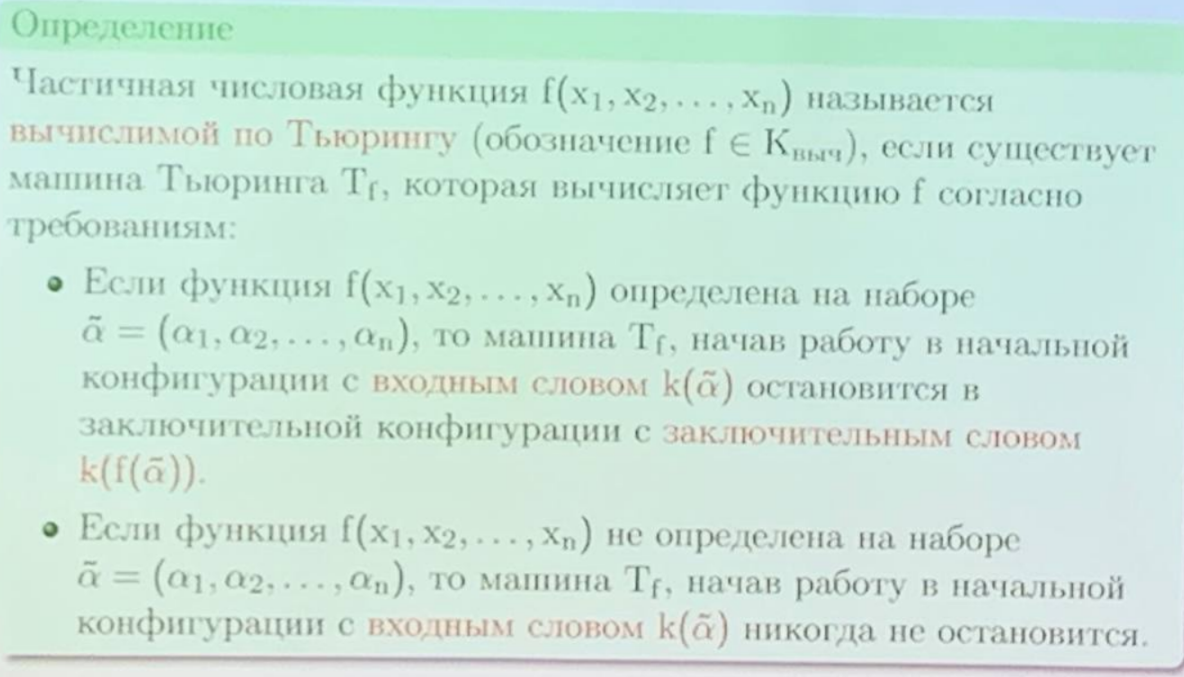
**19. Схемы машин Тьюринга.**



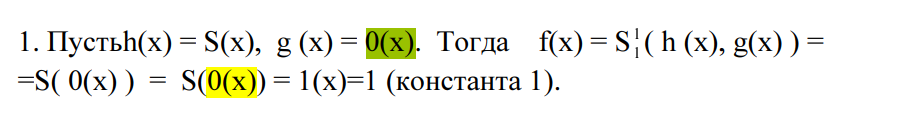
**20. Кодирование чисел и наборов чисел для машин Тьюринга.**



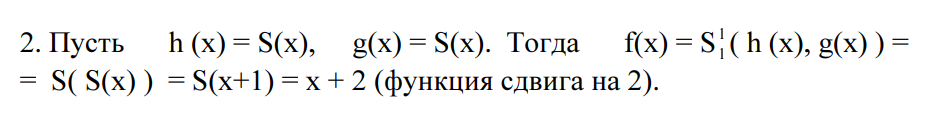
**21. Вычислимость по Тьюрингу.**



**22. Программа вычисления 0(x).**



**23. Программа для S(x).**



**24. Программа вычисления I₂(x, y, z).**

|  |  |  |
| --- | --- | --- |
| I2(x, y, z) | 0 | 1 |
| q1 | q20R | q10R |
| q2 | q30R | q21R |
| q3 | q40R | q30R |
| q4 | q40L | q51L |
| q5 | q00L | q51L |

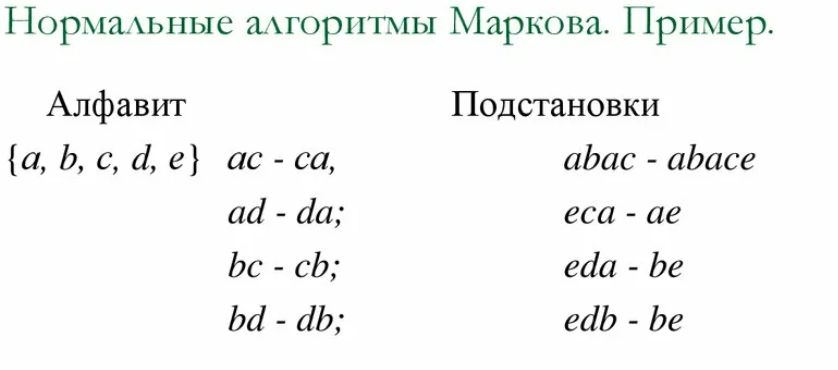
**25. Нормальный алгоритм Маркова: алфавит, подстановка, правила очередности применения подстановок.**

Абстрактным алфавитом называется любая конечная совокупность объектов, называемых буквами или символами данного алфавита. Алфавит, как любое конечное множество, задается перечислением его элементов, т.е. символов.

Пусть Р и Q некоторые слова в заданном алфавите. Тогда выражение Р → Q означает обозначает подстановку т.е. замену слова Р на слово Q.

Правила очередности применения подстановок – Пусть R0 – текущее слово, тогда просматриваем систему подстановок в заданном порядке – с самого начала и находим первую применимую к слову Rp, применяем и получаем новое слово Rp+1. Повторяем то же действие для слова Rp+1.

**26. Пример алгоритма Маркова.**



**27. Тезисы Черча, Тьюринга, Маркова.**

Тезис Черча – Частичная числовая функция является эффективно вычислимой тогда и только тогда, когда она частично-рекурсивна, т.е. Кчр = Кэф.выч

Тезис Тьюринга – Для любого интуитивного алгоритма существует машина Тьюринга, которая его реализует.

Тезис Маркова (принцип нормализации) – Для любого алгоритма в алфавите А существует нормальный алгоритм Маркова, ему эквивалентный.

**28. Временная сложность в наихудшем случае и асимптотическая сложность алгоритма.**

Временная сложность алгоритма (в худшем случае) - это функция размера входных и выходных данных, равная максимальному количеству элементарных операций, алгоритма.

Асимптотическая сложность (асимптотическое описание временной сложности) - оценка скорости роста времени работы алгоритмов, предназначенных для решения одной и той же задачи, при больших объемах входных данных.

**29. Классы P, NP.**

