Оглавление

[1. Критерий полноты системы функций алгебры логики 2](#_Toc294538578)

[2. Проблема полноты в k-значной логике. Алгоритм распознавания Полноты 2](#_Toc294538579)

[3. Ограниченно-детерминированные функции. Операции суперпозиции и обратной связи над ними. Конечная порожденность класса о.-д. функции относительно этих операций. 3](#_Toc294538580)

[4. Алфавитное кодирование. Алгоритм распознавания однозначности алфавитного кодирования 3](#_Toc294538581)

[5. Эквивалентные преобразования в функциональных системах. Конечные полные системы тождеств для ФАЛ и СФЭ. 3](#_Toc294538582)

[6. Сокращенные, тупиковые, минимальные дизъюнктивные нормальные формы (д.н.ф.), алгоритмы их построения. Оценки сложности д.н.ф. 4](#_Toc294538583)

[7. Метод Лупанова для синтеза схем из функциональных элементов 5](#_Toc294538584)

[8. Сложность алгоритмов. Классы P и NP. Теорема об NP-полноте задачи о выполнимости. 7](#_Toc294538585)

[9. Независимые случайные величины. Критерий независимости случайных величин. 7](#_Toc294538586)

[10. Моменты случайных величин. Свойства математических ожиданий и дисперсий 8](#_Toc294538587)

[11. Центральная предельная теорема 9](#_Toc294538588)

[12. Точечные и интервальные оценки неизвестных параметров распределения. Свойства точечных оценок (несмещенность, состоятельность, эффективность, оптимальность). Два метода построения точечных оценок (метод максимального правдоподобия, метод моментов) 9](#_Toc294538589)

[13. Основные понятия о проверке статистических гипотез. Лемма Неймана-Пирсона 11](#_Toc294538590)

[14. Доверительные интервалы для параметров нормального распределения 12](#_Toc294538591)

[15. Виды сходимости последовательностей случайных величин 13](#_Toc294538592)

[16. Основная теорема матричных игр 13](#_Toc294538593)

[17. Иерархические игры и их решение 14](#_Toc294538594)

[18. Теорема Гермейера о решении игры 15](#_Toc294538595)

[19. Принцип уравнивания в задаче оптимального распределения ресурсов 15](#_Toc294538596)

[20. Выпуклые множества и выпуклые функции. Необходимое и достаточное условие оптимальности в общей задаче минимизации 16](#_Toc294538597)

[21. Задачи линейного программирования: прямая и двойственная, их свойства. Основная идея симплекс-метода 17](#_Toc294538598)

[22. Описание статистической модели Леонтьева. Условие продуктивности. 19](#_Toc294538599)

[23. Модель Курно 20](#_Toc294538600)

[24. Постановка задачи оптимального управления. Понятие о задаче синтеза 21](#_Toc294538601)

[25. Множество достижимости линейной управляемой системы. Его опорная функция. 22](#_Toc294538602)

[26. Управляемость и локальная управляемость линейных систем. 23](#_Toc294538603)

[27. Принцип Максимума Понтрягина для линейной задачи быстродействия. 24](#_Toc294538604)

[28. Уравнения в вариациях. Построение конуса касательных направлений к множеству достижимости. 25](#_Toc294538605)

[29. Принцип максимума Понтрягина для задачи оптимального управления с интегральным функционалом 26](#_Toc294538606)

[30. Понятие о методе динамического программирования 27](#_Toc294538607)

# Критерий полноты системы функций алгебры логики

**Опр**. Система функций из называется функционально полной, если любая булева функция может быть записана в виде формулы через функции этой системы.

Обозначим через класс всех булевых функций, сохраняющих константу 0, т.е. функций, для которых выполнено равенство *f* (0,...,0) = 0.

Обозначим через класс всех булевых функций, сохраняющих константу 1, т.е. функций, для которых выполнено равенство *f* (1,...,1) = 1.

Обозначим через ***S***класс всех самодвойственных функций, т.е. функций *f* из 2 *P*, таких что т.е. .

**Опр**. Функция *f* называется монотонной, если ∀ двух наборов α и β: α β , имеет место равенство *f* (α ) ≤ *f* (β) .

Обозначим через ***M***множество всех монотонных функций.

**Теорема** (о функциональной полноте). Для того чтобы система функций была полной, необходимо и достаточно, чтобы она целиком не содержалась ни в одном из пяти замкнутых классов , *S*, *M* и *L* .

# Проблема полноты в k-значной логике. Алгоритм распознавания Полноты

**Опр**. Система *B* функций из называется функционально полной, если любая функция из может быть записана в виде формулы через функции этой системы.

**Теорема**. Система является полной в , где ,

**Теорема**. Существует алгоритм для распознавания полноты.

**Доказательство**. Построим по индукции последовательность множеств функций от двух переменных .

*Базис индукции*.

Положим , где – пустое множество.

*Индуктивный переход*.

Пусть уже построены множества , построим :

Выпишем функции, входящие в ( r ≥ 0 ): ( при ), и

для каждого рассмотрим всевозможные формулы вида , где есть либо функция (), либо .

Просматривая формулы, мы получим функции, не вошедшие в : .

Положим .

Очевидно, что .

Из построения ясно, что если , то , т.е. цепочка множеств

стабилизируется.

Обозначим через минимальный номер множества, начиная с которого наступает стабилизация. Тогда цепочка множеств строго возрастает.

Момент стабилизации может быть обнаружен через ограниченное число шагов.

Рассмотрим множество .

Возможны два случая.

1. содержит все функции от двух переменных и, значит, содержит . Тогда исходная система полна.
2. не содержит всех функций от двух переменных. Поскольку в этом случае замыкание , то замыкание не содержит всех функций от переменных . Следовательно, не полна.

*Алгоритм: строим классы до момента стабилизации и рассматриваем класс , по которому и определяем, имеет место полнота для B или нет.*

# 3. Ограниченно-детерминированные функции. Операции суперпозиции и обратной связи над ними. Конечная порожденность класса о.-д. функции относительно этих операций.

**Опр**. Функция  называется **детерминированной**, если каково ни было число  и каковы бы ни были последовательности  и  такие, что , , …, , значения  и  функции , где , , представляют собой последовательности, у которых тоже совпадают первые  членов.

**Опр.** Детерминированная функция называется **ограниченно-детерминированной**, если она имеет конечный вес.

Замкнутость класса о-д функций относительно операции суперпозиции.



**Теорема 3**. Класс детерминированных функций замкнут относительно операции суперпозиции.

**Теорема 4**. Класс о-д функций замкнут относительно операции суперпозиции.

Операция введения обратной связи. Замкнутость класса о-д функций относительно операции введения обратной связи.

**Опр**. Детерминированная функция  зависит от переменной  с запаздыванием, если для любых входных последовательностей ,  и любого момента времени  значение , где , полностью определяется значениями первых  членов последовательностей  и значениями  членов последовательности .



Пусть  зависит от  с запаздыванием. Введем обратную связь по переменным  и .

Тогда и 

**Теорема 5**. Класс о-д функций замкнут относительно операции обратной связи.

**Теорема 6**. Пусть для системы о-д функций , где , возможно введение обратных связей и в порядке , , и в порядке ,. Тогда результаты применения операции обратной связи совпадают.

Теорема7. Класс  замкнут относительно операций С и О.

# 4. Алфавитное кодирование. Алгоритм распознавания однозначности алфавитного кодирования

**Опр.** Пусть *A* = {*a*1, *a*2, …, *ar*} — *исходный алфавит*, *B* = {*b*1, *b*2, …, *bm*} — *кодирующий алфавит,* *A*\* = ∅∪ *A*∪ *A*2 ∪ *A*3 ∪…*AN* ∪… , *B*\* = ∅∪ *B* ∪ *B*2 ∪ *B*3 ∪…*BN* ∪… (множества всех возможных слов в алфавитах).

Тогда *алфавитным кодированием A*\* → *B*\* назовём отображение ϕ : *A* → *B*\* такое, что *ai* → *Bi*.

Множество {*B*1, *B*2, …, *Br*} при этом называется множеством *кодовых слов* (или просто *кодом*). При этом .

Пусть W — максимальное число кодовых слов, которые «помещаются» подряд внутри кодового слова. Пусть — длина слова и .

**Собственно алгоритм** определения однозначности некоторого кодирования ϕ, следующий из теоремы: кодируются все слова, длины не более . Если все коды получились уникальными, значит и для всех более длинных слов нарушений однозначности не будет.

# 5. Эквивалентные преобразования в функциональных системах. Конечные полные системы тождеств для ФАЛ и СФЭ.

**Опр.** Функциональная система: пара (Σ, *f* ), где *f* - некоторая функция алгебры логики, Σ - способ реализации этой функции. Σ может быть контактной схемой (КС), схемой из функциональных элементов (СФЭ) или формулой в базисе (∨,&,¬). Символом *f* Σ будем обозначать схему, реализующую нек. функцию *f* .

**Опр.** Будем говорить, что *f* Σ эквивалентно *f* ′ Σ′ ( *f* Σ ~ *f* ′ Σ′ ), если *f* = *f* ′ . Аналогично *V* = (Σ, *f* ) ~ *V* ′ = (Σ′, *f* ′) если *f* = *f* ′ .

**Опр.** Подстановка это корректная замена подсхемы Σ′ на подсхему Σ′′ . Подстановка является эквивалентной, если Σ′ ~Σ′′ .

Пусть от схемы 1 Σ с помощью эквивалентной подстановки переходим к схеме 2 Σ ( 1 Σ →2Σ), от 2 Σ к 3 Σ , и так далее до *p* Σ . Тогда *p* Σ получена из 1 Σ с помощью эквивалентных преобразований ( *p* Σ ⇒ Σ 1 ). Очевидно, что в таком случае верно и 1 Σ ⇒ Σ *p* . В таком случае можно говорить, что 1 Σ ~ *p* Σ относительно системы тождеств T (эти тождества использовались при эквивалентных подстановках).

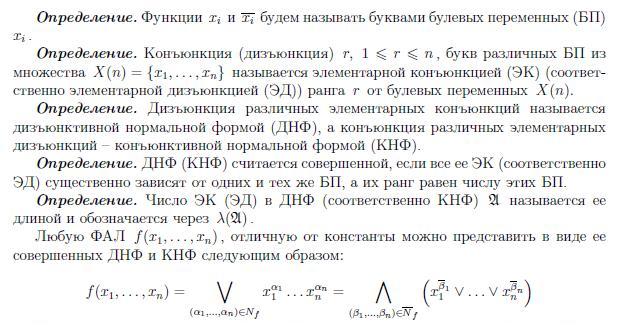
**Опр.** Система тождеств Т полна в рассматриваемом классе функциональных систем К, если ∀ Σ ~ Σ′ , Σ,Σ′∈ *K* ⇒ Σ Σ′.

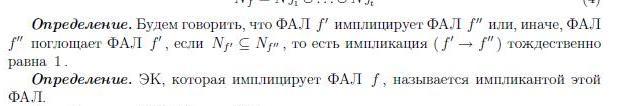
**Теорема.** Для класса формул в базисе = {∨,&,¬} 0 *Б* существует конечная полная система тождеств.

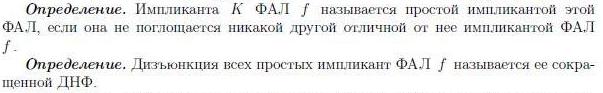
**Опр.** Для схем из функциональных элементов в определении эквивалентности нужно уточнить, что схемы Σ и Σ′ эквивалентны, если у них одинаковые наборы входов, выходов, и все реализованные функции также одинаковы.

**Теорема.** В классе СФЭ существует КПСТ.

# **6. Сокращенные, тупиковые, минимальные дизъюнктивные нормальные формы (д.н.ф.), алгоритмы их построения. Оценки сложности д.н.ф.**

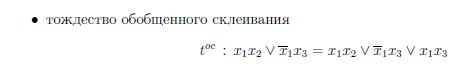




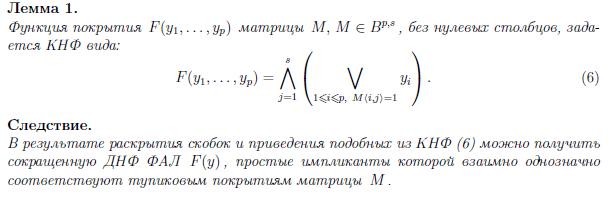


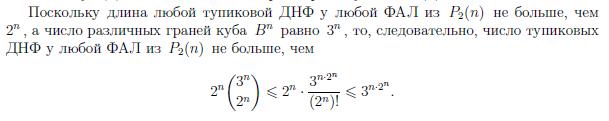
Алгоритм (метод Блейка). Данный алгоритм использует систему тождеств:

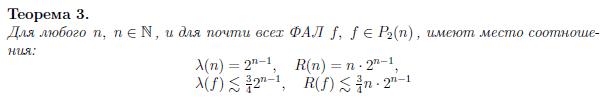




8.JPG







# **7. Метод Лупанова для синтеза схем из функциональных элементов**

**Теорема (О.Б. Лупанов).** Для схем из Ф.Э. в базисе, состоящем из инвертора,

дизъюнктора и конъюнктора, можно построить асимптотически наилучший метод синтеза

и

**Алгоритм:**

1. Зададим произвольную булеву функцию при помощи таблицы размера 2*k* × 2*n*−*k*

Строки этой таблицы нумеруем наборами значений по переменным , столбцы

– по переменным . На пересечении строки и столбца

помещаем значение *f ()*.

Легко видеть, что столбец с номером задает функцию

*f (*, являющуюся компонентой разложения

2. Возьмем целое число *s*, 1 < *s* < 2*k* , и разрежем таблицу на полосы шириной *s.*

Занумеруем сверху вниз полученные полосы числами *1,2,…,p*, где .

3. рассмотрим полосу с номером *i* и строками . Эта полоса распадается на короткие столбцы высоты *s*. Поэтому число *t(i)* сортов коротких столбцов будет не более . Произведем нумерацию этих сортов числами *1,2,…,t(i).*

4. Пусть - столбец *j*-го сорта. Обозначим через булеву функцию,

определяемую этим коротким столбцом:

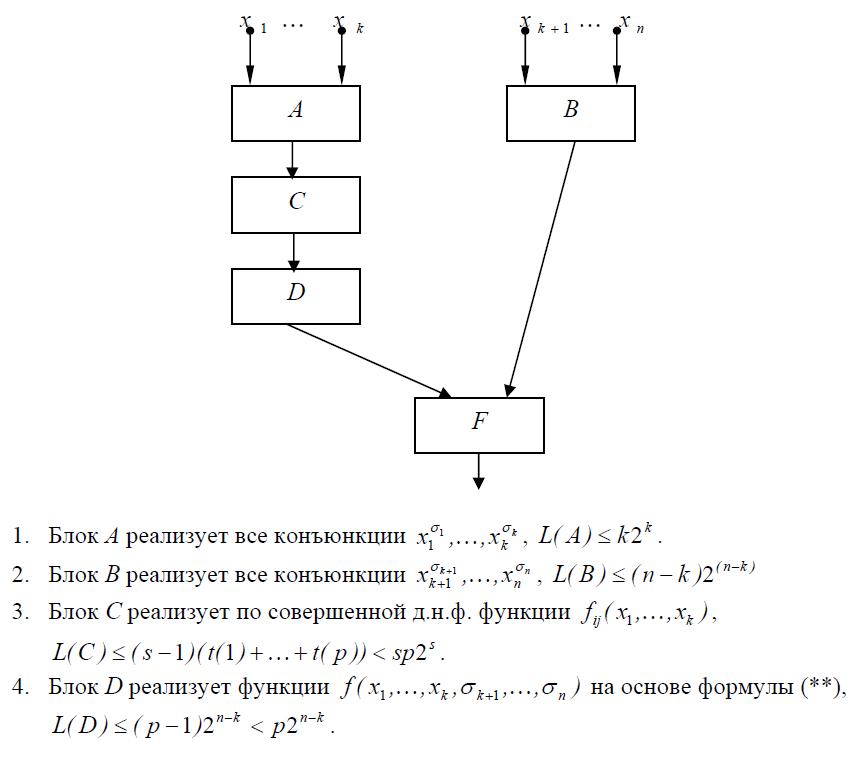
Столбец с номером разрезается полосами на *p* коротких столбцов.

Поэтому

*f (*

где- номер сорта соответствующего короткого столбца, принадлежащего *i*-ойполосе.

Теперь перейдем к описанию схемы Σ . Ее мы получим в виде соединения отдельных блоков.



5. Блок *F* реализует функцию исходя из ее разложения (\*),

*L( F )* ≤ 2*n*−*k* + 2*n*−*k* −1 < 2 ⋅ 2*n*−*k* .

Поэтому

В силу произвольности функцииотсюда следует, что

# 8. Сложность алгоритмов. Классы P и NP. Теорема об NP-полноте задачи о выполнимости.

**Опр**. МТ называется *детерминированной*, если для каждой пары вида *(a,q)*, *a*∈ *A,q*∈*Q* в программе МТ присутствует не более одной команды вида *aq* →*a' q' d* .

**Опр**. Если существует пара *(a,q)*, для которой существует ≥ 2 команд, начинающихся с *aq*, то МТ называется *недетерминированной*.

**Обз**. Пусть *A* – конечный алфавит. Через *A\** обозначим множество всех конечных слов в алфавите *A*.

**Опр**. Произвольное подмножество *L*∈ *A\** называется *языком* в алфавите *A*.

**Обз**. Через P обозначим класс языков, распознающихся МТ за полиномиальное время.

**Обз**. Класс языков, распознающихся НМТ за полиномиальное время, обозначим через NP.

**Опр**. Пусть *П* – множество отображений вида *f : A\** → *A\** , вычисляемых МТ за полиномиальное время. Пусть *L* и *K* – языки. Говорят, что *L полиномиально сводится* к *K* ( *L K* ), если существует *f* ∈ *П : f ( w)*∈ *K* ⇔ *w*∈ *L* .

**Опр**. Язык называется *NP-полным*, если:

1) *L*∈ *NP;*

2) ∀*K* ∈ *NP* ⇒ *K*  *L.*

**Язык Выполнимость (ВЫП):**

*A= {(, ),&, , , ,i=1,2,.. }*

*w*∈ *ВЫП* ⇔ *w***-**КНФ, не равная тождественно 0.

**Теорема Кука.** Если *L*∈ *NP* , то *L*  *ВЫП* .

**Следствие теоремы Кука.** Язык ВЫП – NP-полный.

**Язык k-ВЫП.**

ВХОД: k-КНФ *F(*;

СВОЙСТВО: выполнимость.

k-КНФ – конъюнкция скобок, каждая из которых является дизъюнкцией не более k

букв.

**Теорема.** Язык *ВЫП* ∈ *NP* .

# 9. Независимые случайные величины. Критерий независимости случайных величин.

*Определение.* Случайные величины называются независимыми, если

∀ , где **B** – борелевская σ -алгебра множеств на числовой прямой **R,**

**Теорема (критерий независимости случайных величин).** Для того, чтобы были взаимно независимыми, необходимо и достаточно, чтобы при любом выборе *n* интервалов *[ , ),… ,[ , )* на числовой прямой

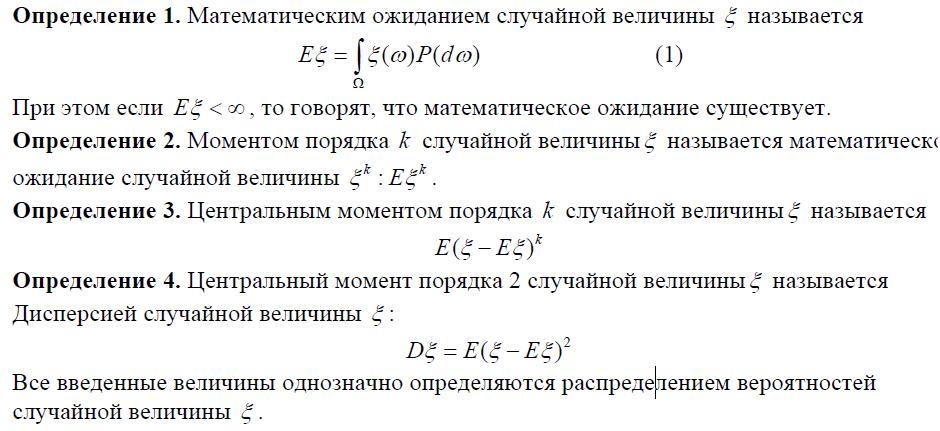
*P{ w: (w),… ,(w)} = P{ w: (w)}… P{ w:(w)}*

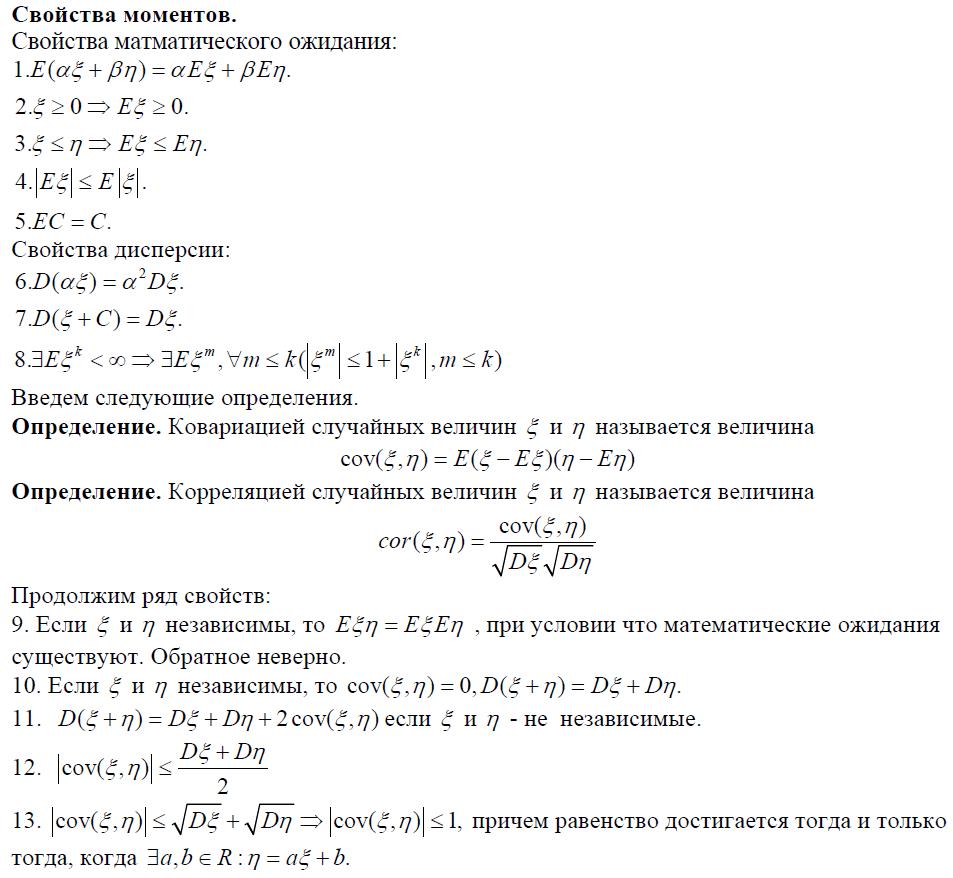
**Свойства независимых случайных величин:**

Для независимых случайных величин *X* и *Y*

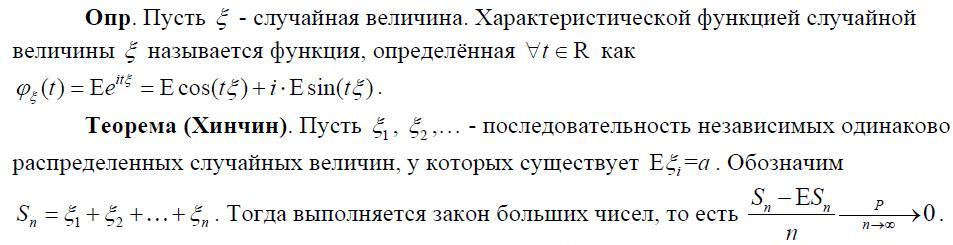
*E( XY )= EX EY*, *D( X+ Y )= DX+ DY*

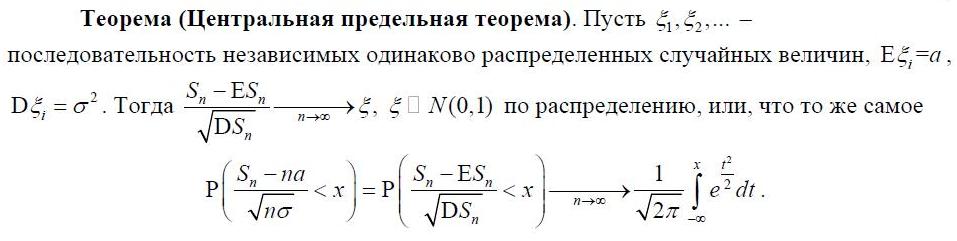
# 10. Моменты случайных величин. Свойства математических ожиданий и дисперсий



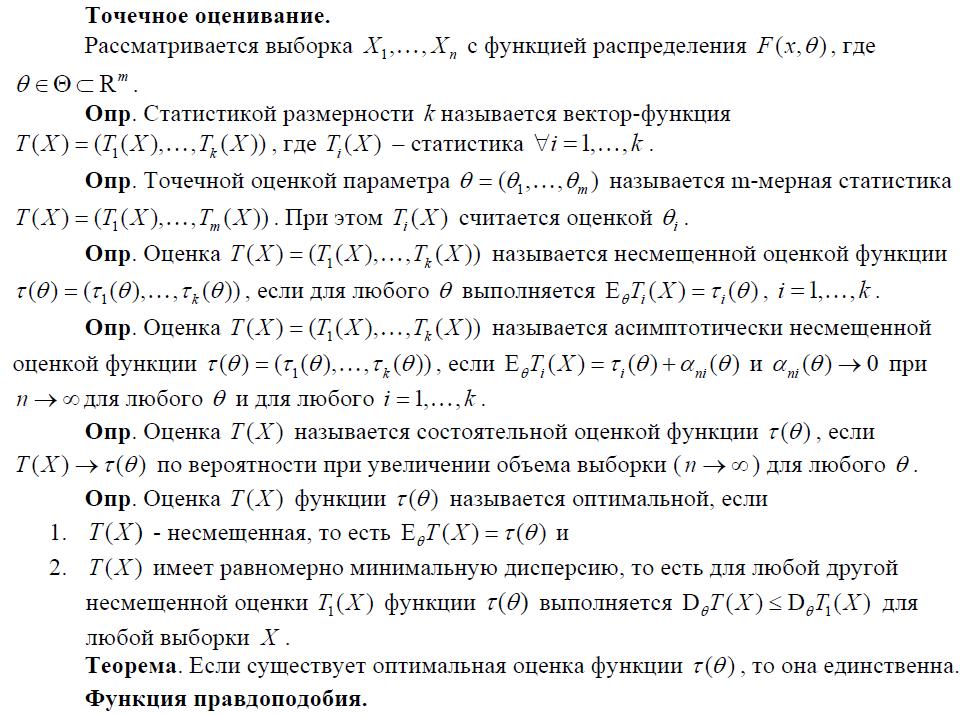


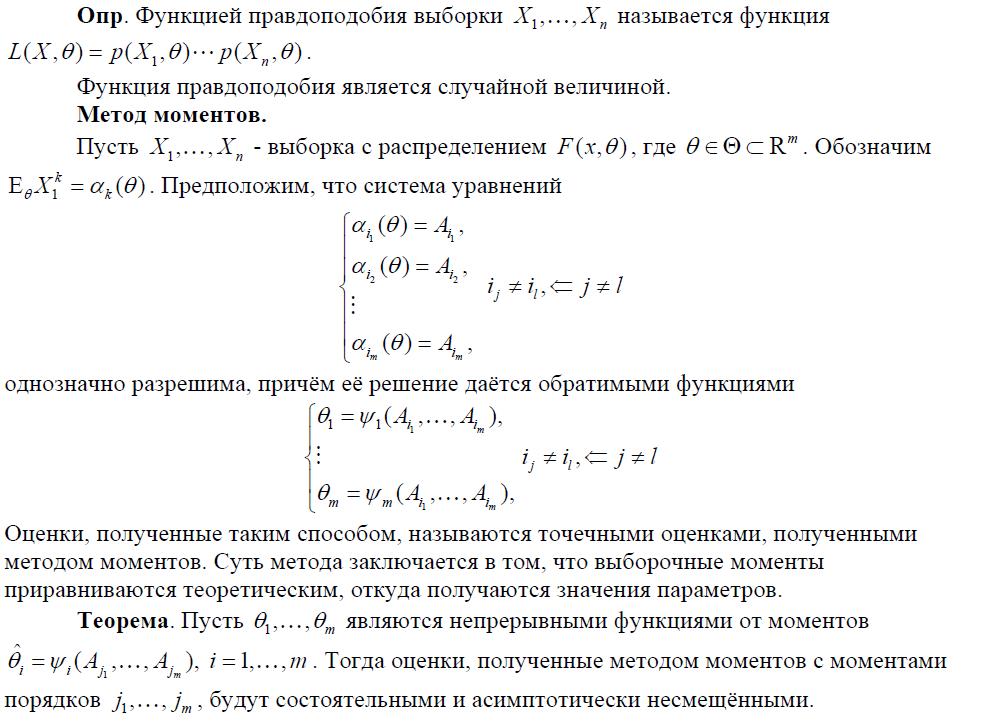
# 11. Центральная предельная теорема





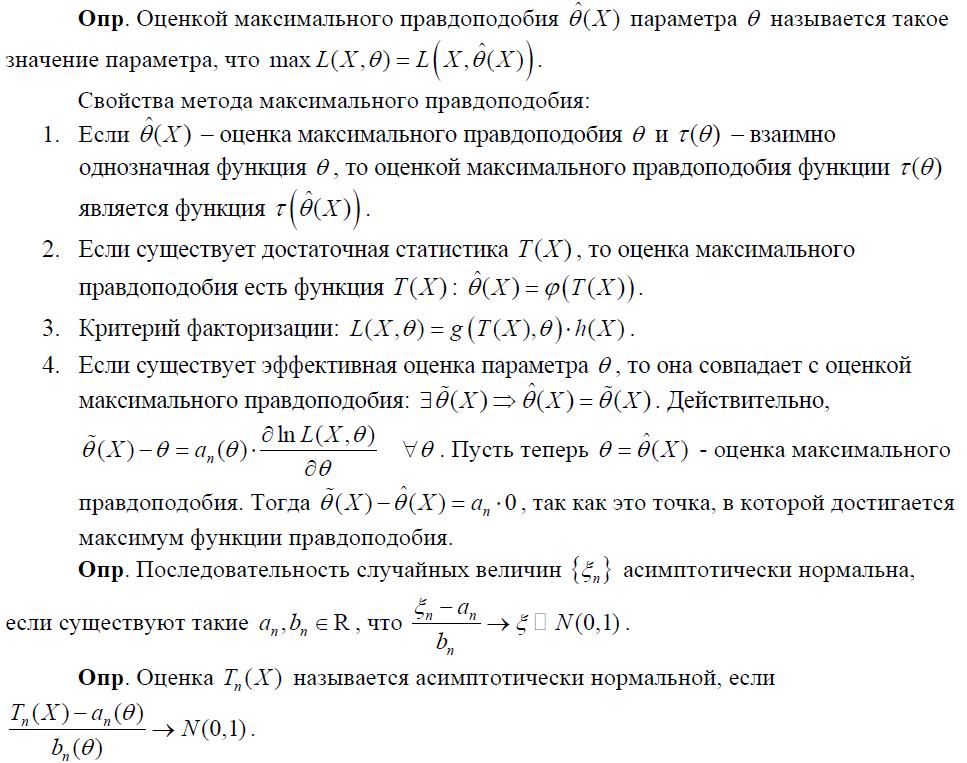
# 12. Точечные и интервальные оценки неизвестных параметров распределения. Свойства точечных оценок (несмещенность, состоятельность, эффективность, оптимальность). Два метода построения точечных оценок (метод максимального правдоподобия, метод моментов)

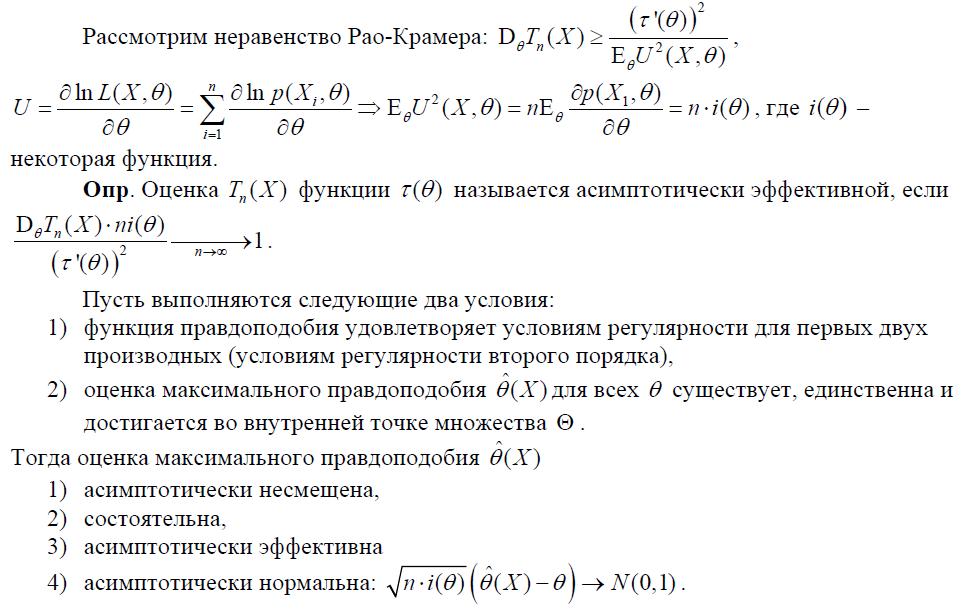




**Метод максимального правдоподобия.**

Пусть *L*(*X* ,θ ) - функция правдоподобия выборки *X* .

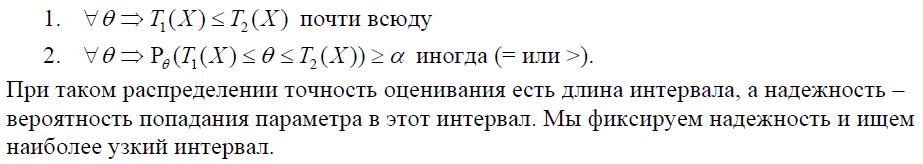
****

****

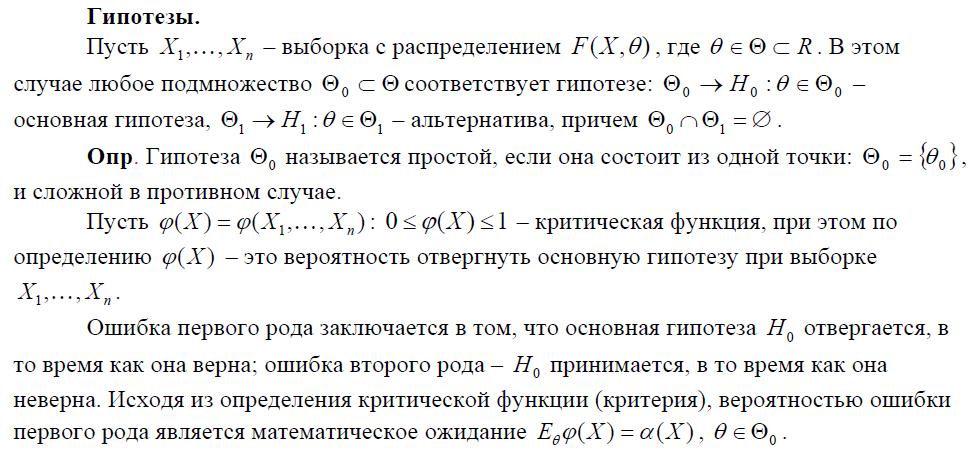
**Интервальное оценивание.**

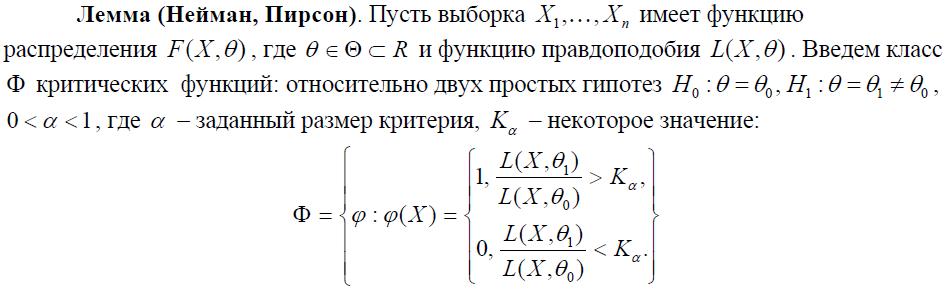
**Опр**. Доверительным интервалом с коэффициентом доверия 0 ≤α ≤1 называется

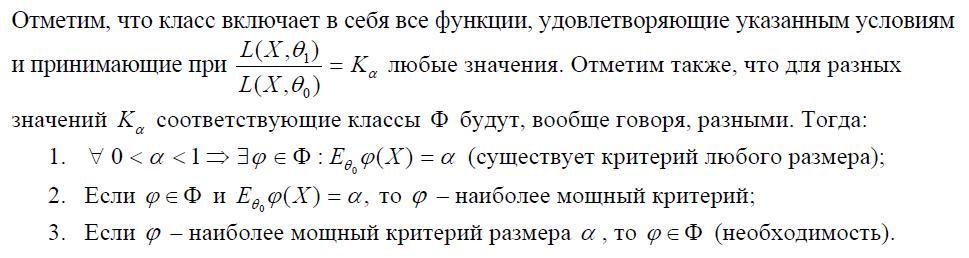
совокупность двух статистик ((*X* ),(*X* )) таких, что

****

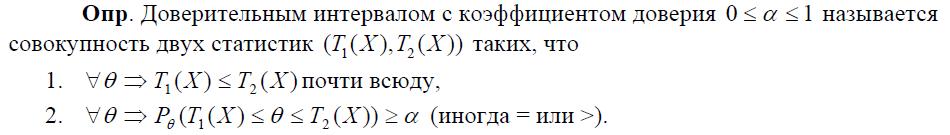
# 13. Основные понятия о проверке статистических гипотез. Лемма Неймана-Пирсона

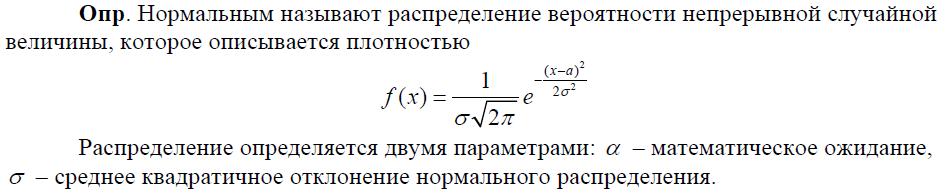


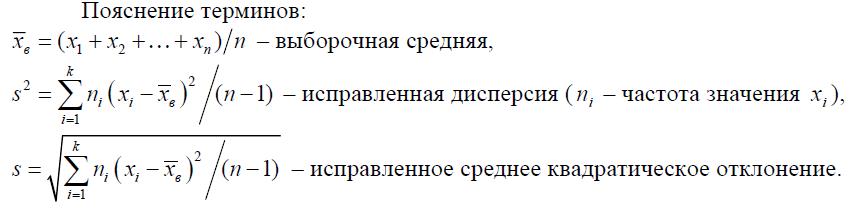




# 14. Доверительные интервалы для параметров нормального распределения

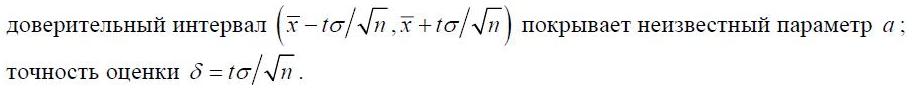




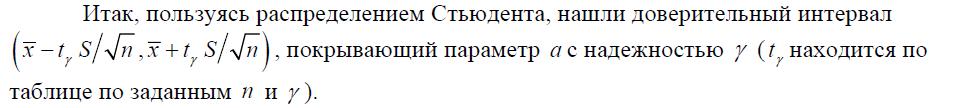




**Доверительные интервалы для оценки математического ожидания нормального распределения при известном σ:**

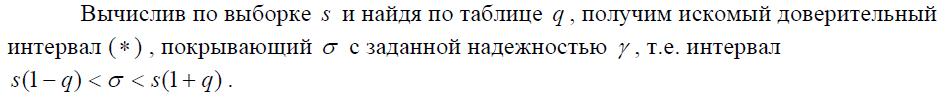


**Доверительные интервалы для оценки математического ожидания нормального распределения при неизвестном σ.**

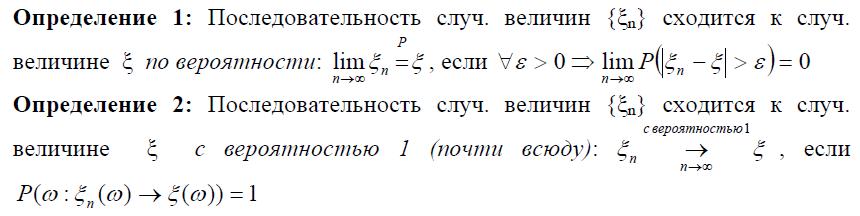
****

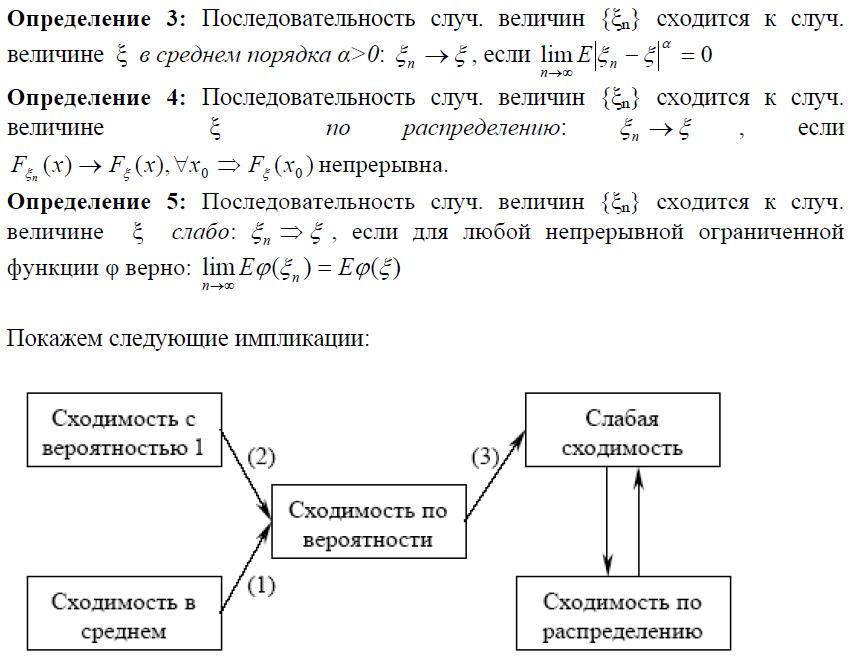
**Доверительные интервалы для оценки среднего квадратического отклонения**

**σ нормального распределения.**

****

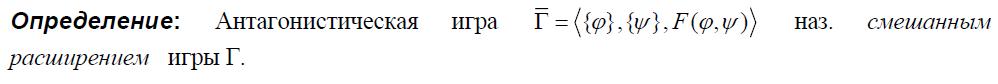
# 15. Виды сходимости последовательностей случайных величин

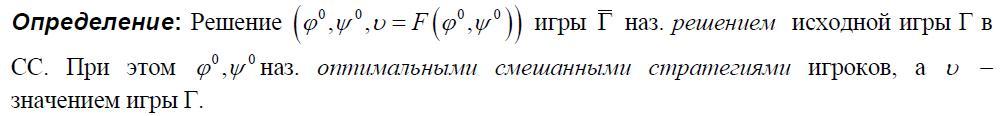




# 16. Основная теорема матричных игр

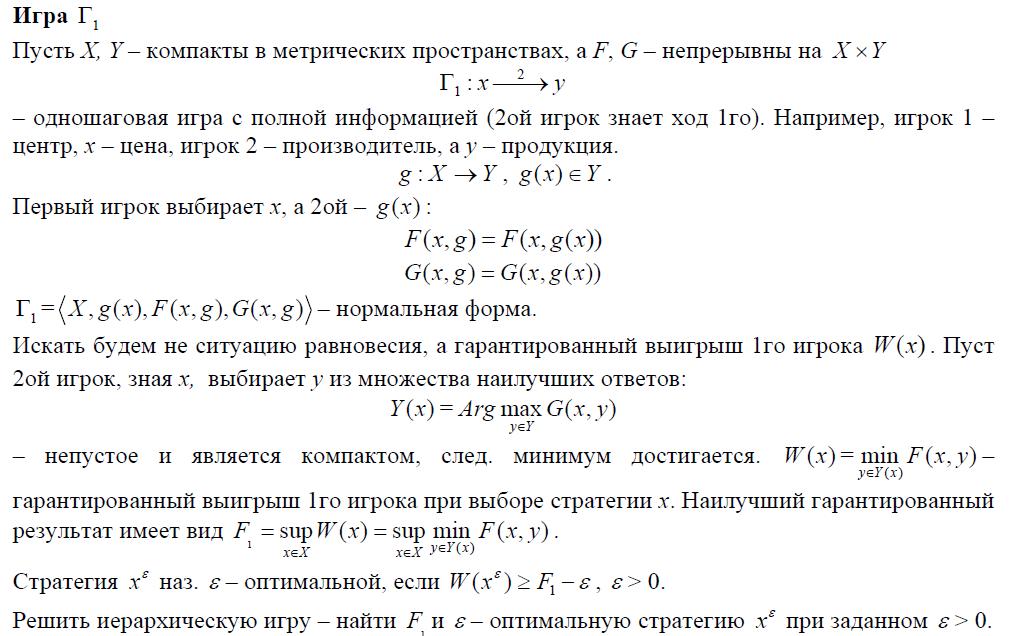
**Определение**: Смешанной стратегией (СС) 1го игрока игры Г называется вероятностное распределение ϕ на множестве стратегий X.

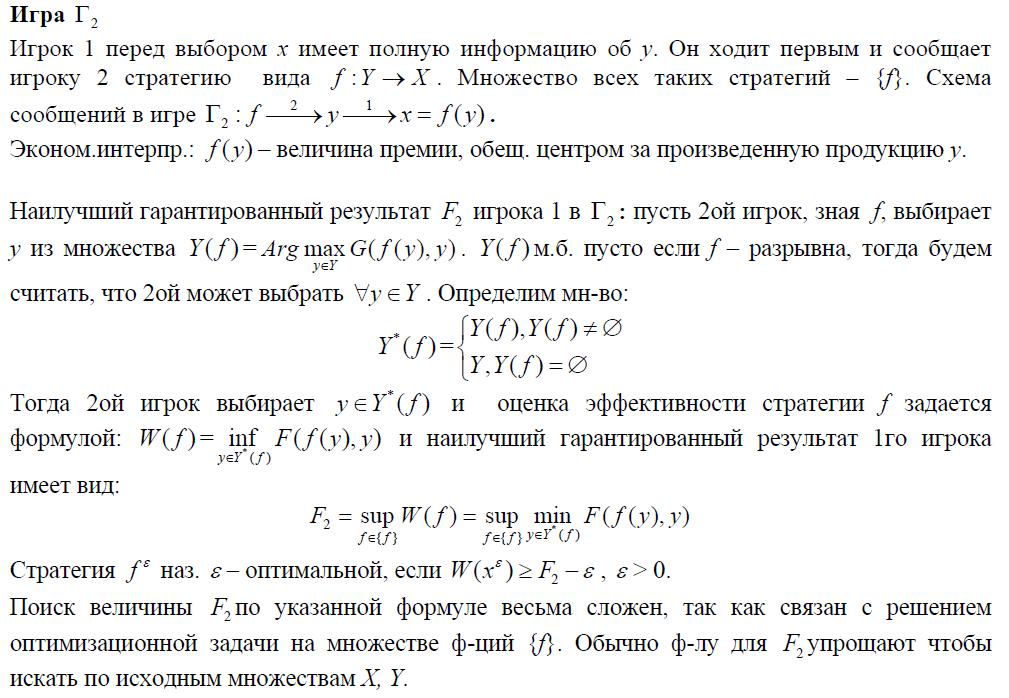


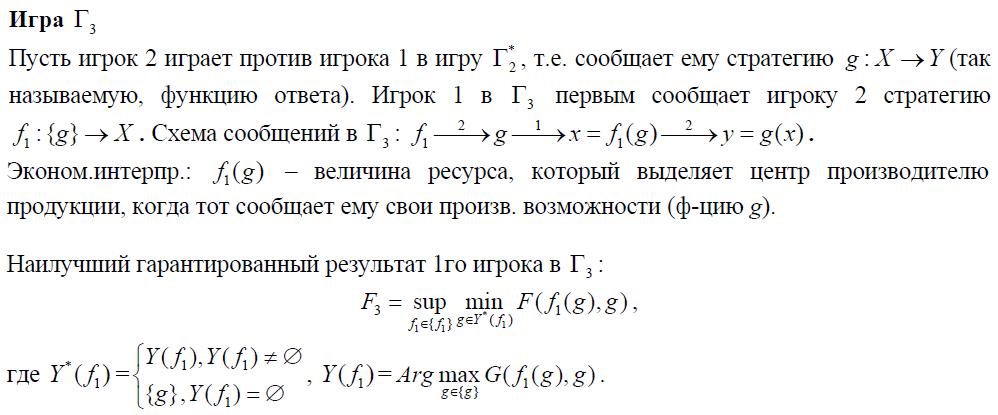


38.JPG

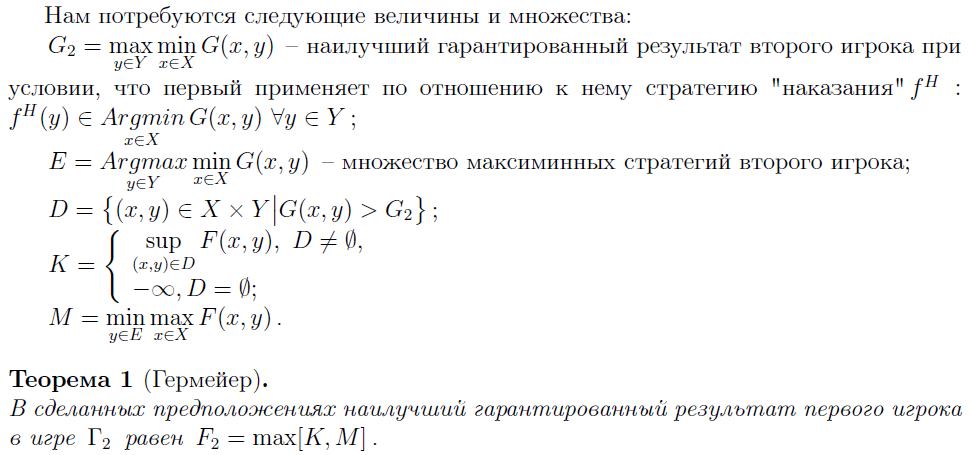
# 17. Иерархические игры и их решение



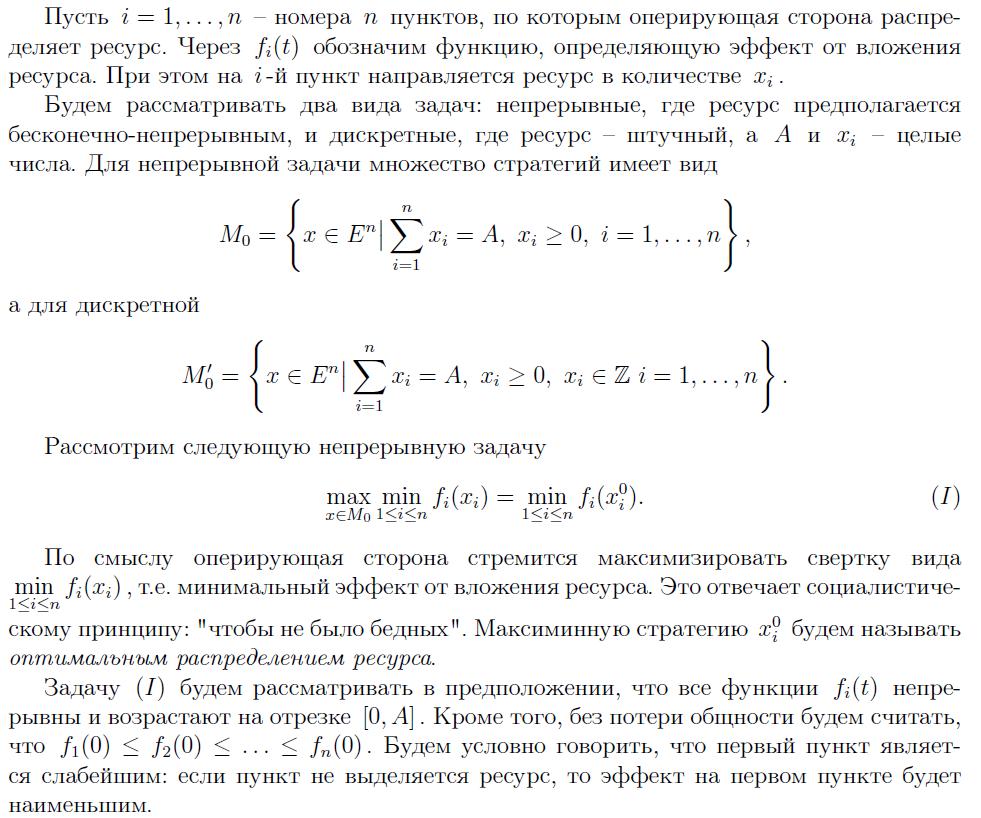


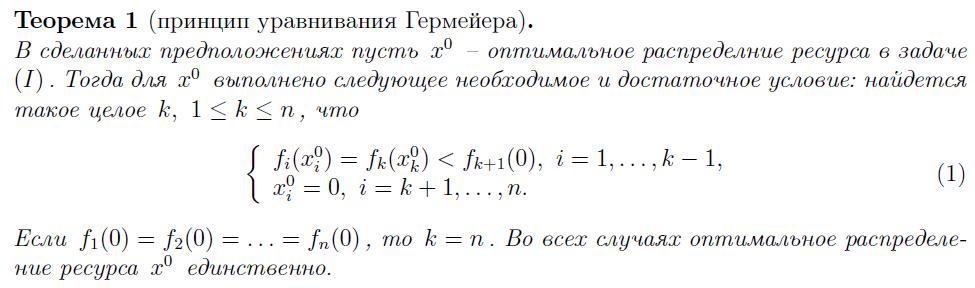


# 18. Теорема Гермейера о решении игры

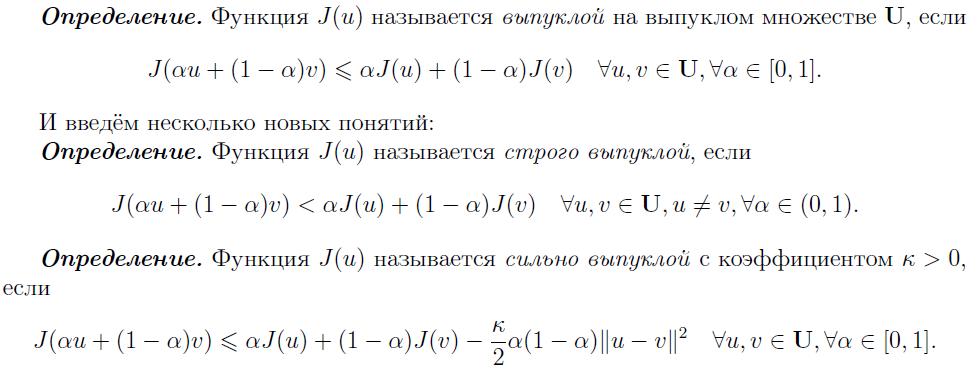


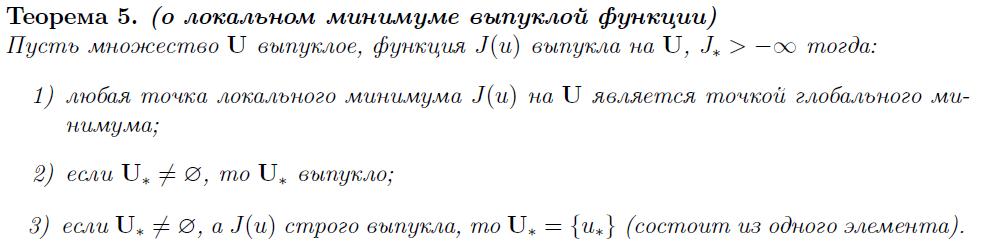
# 19. Принцип уравнивания в задаче оптимального распределения ресурсов

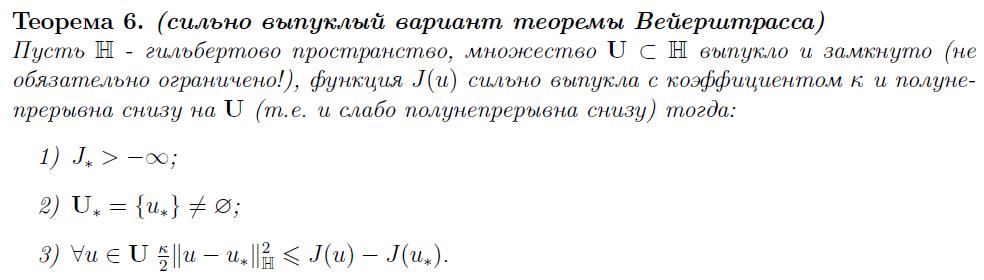


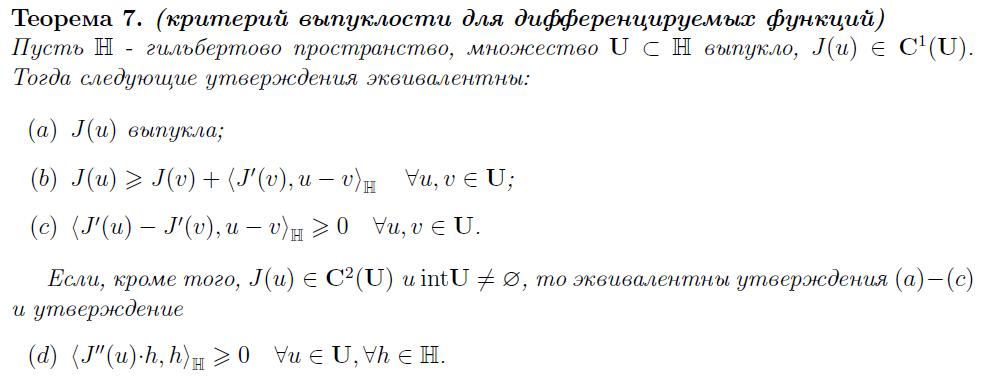


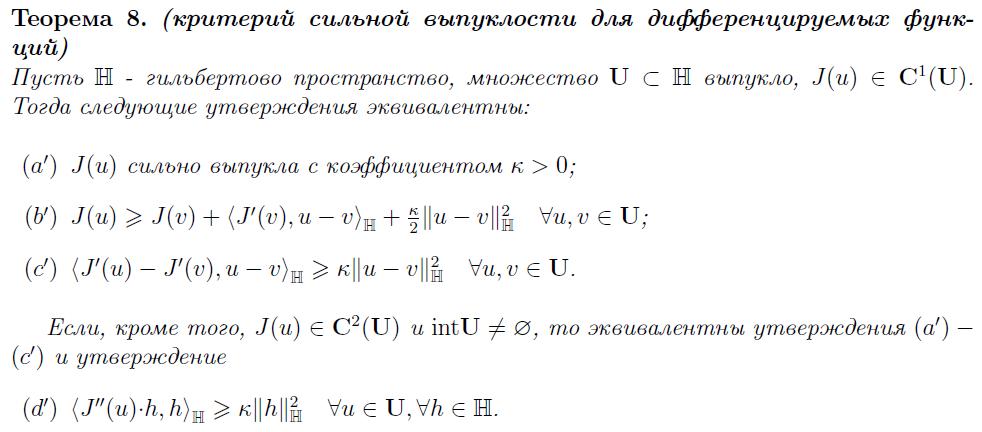
# 20. Выпуклые множества и выпуклые функции. Необходимое и достаточное условие оптимальности в общей задаче минимизации

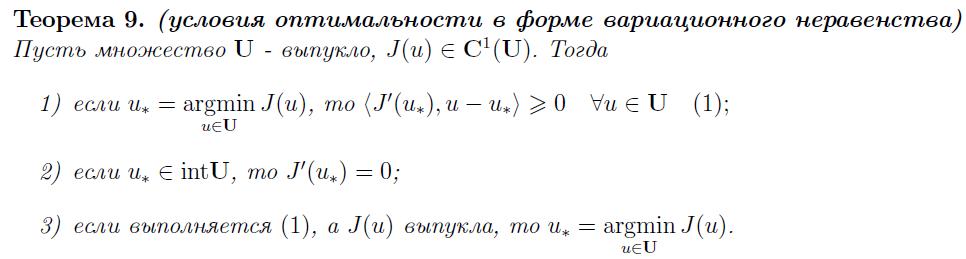




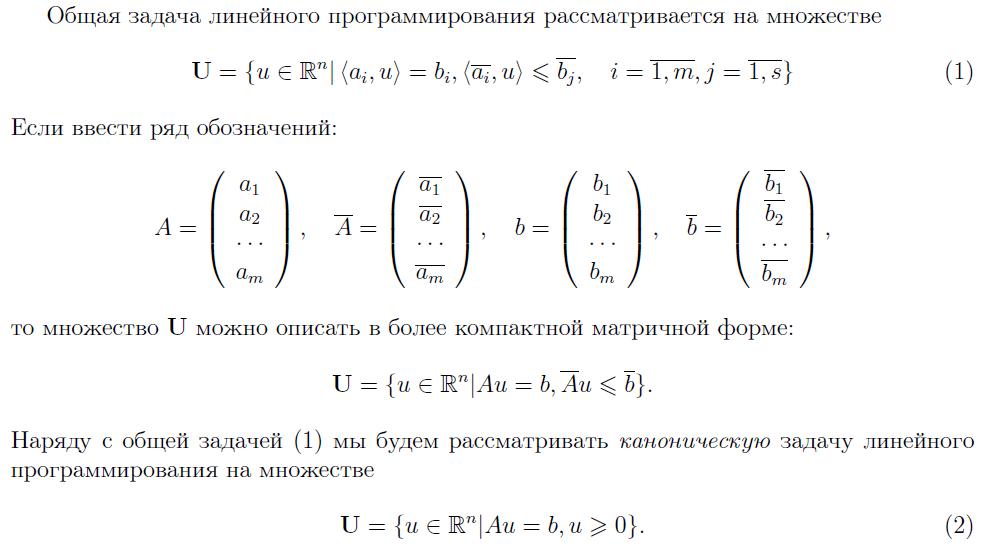


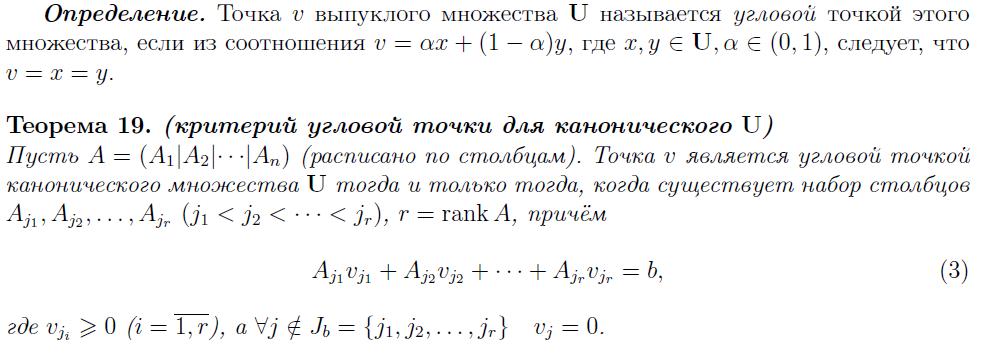


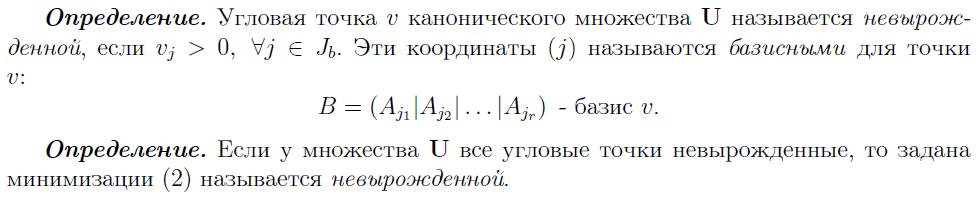


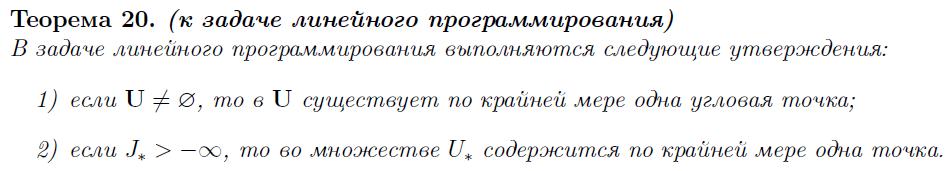


# 21. Задачи линейного программирования: прямая и двойственная, их свойства. Основная идея симплекс-метода









**Симплекс-метод.**

Симплекс-метод — алгоритм решения оптимизационной задачи линейного программирования путём перебора вершин выпуклого многогранника в многомерном пространстве.

Задача линейного программирования состоит в том, что необходимо максимизировать или минимизировать некоторый линейный функционал на многомерном пространстве при заданных линейных ограничениях.

Заметим, что каждое из линейных неравенств на переменные ограничивает полупространство в соответствующем линейном пространстве. В результате все неравенства ограничивают некоторый многогранник (возможно, бесконечный), называемый также полиэдральным комплексом. Уравнение W(x) = c, где W(x) — максимизируемый (или минимизируемый) линейный функционал, порождает гиперплоскость L(c). Зависимость от c порождает семейство параллельных гиперплоскостей. Тогда экстремальная задача приобретает следующую формулировку — требуется найти такое наибольшее c, что гиперплоскость L(c) пересекает многогранник хотя бы в одной точке. Заметим, что пересечение оптимальной гиперплоскости и многогранника будет содержать хотя бы одну вершину, причём, их будет более одной, если пересечение содержит ребро или k-мерную грань. Поэтому максимум функционала можно искать в вершинах многогранника. Принцип симплекс-метода состоит в том, что выбирается одна из вершин многогранника, после чего начинается движение по его рёбрам от вершины к вершине в сторону увеличения значения функционала. Когда переход по ребру из текущей вершины в другую вершину с более высоким значением функционала невозможен, считается, что оптимальное значение c найдено.

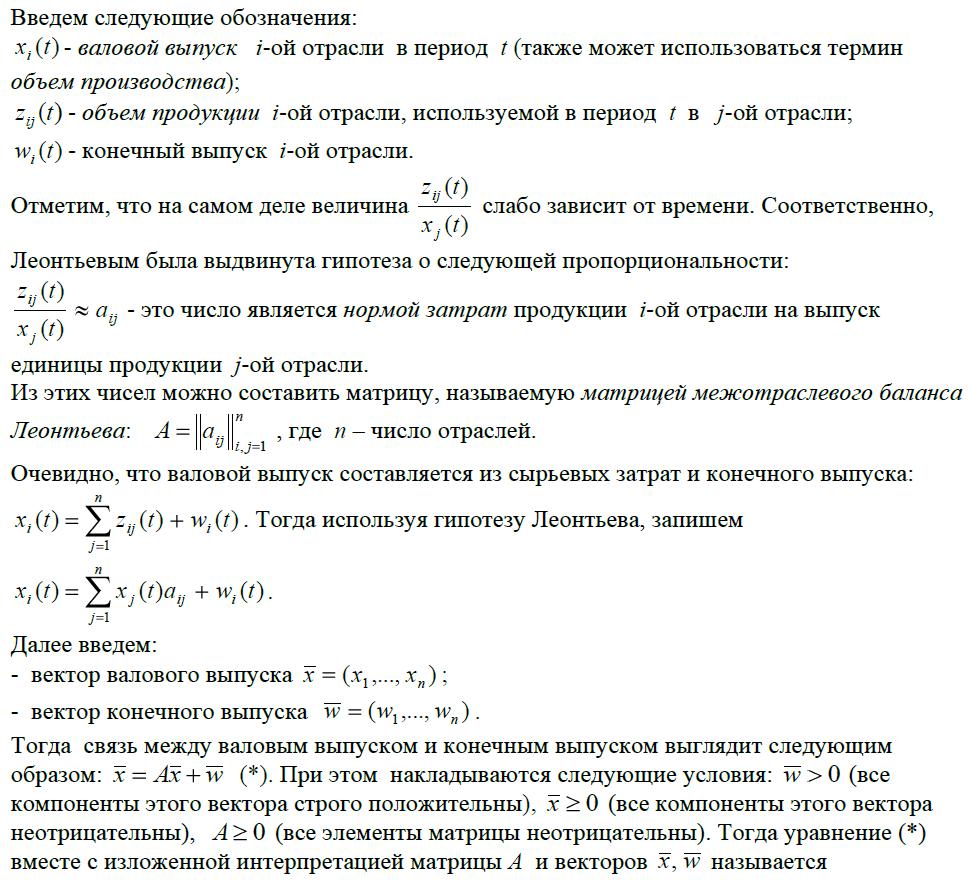
Последовательность вычислений симплекс-методом можно разделить на две основные фазы:

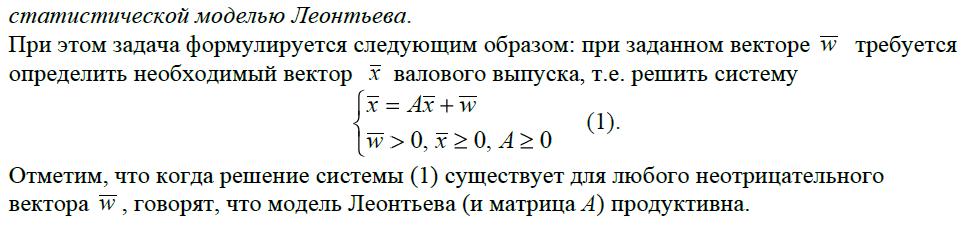
1. нахождение исходной вершины множества допустимых решений,

2. последовательный переход от одной вершины к другой, ведущий к оптимизации значения целевой функции.

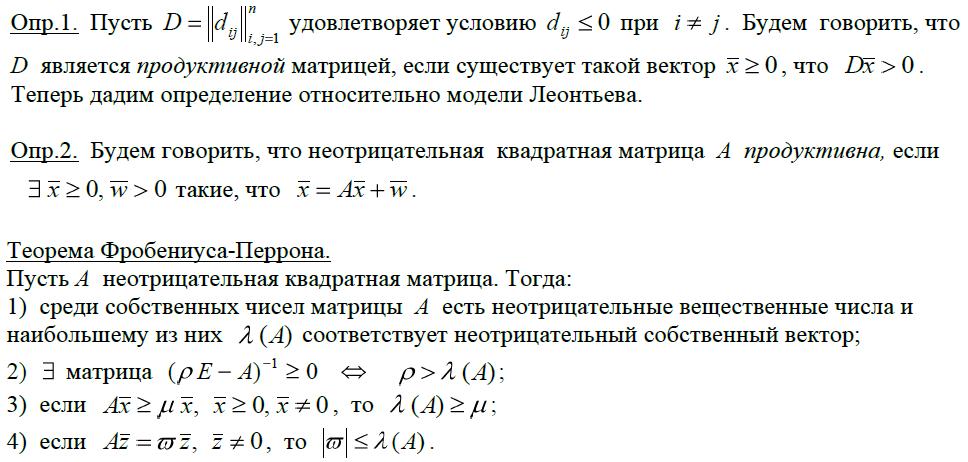
При этом в некоторых случаях исходное решение очевидно или его определение не требует сложных вычислений, например, когда все ограничения представлены неравенствами вида «меньше или равно» (тогда нулевой вектор совершенно точно является допустимым решением, хотя и, скорее всего, далеко не самым оптимальным). В таких задачах первую фазу симплекс-метода можно вообще не проводить. Симплекс-метод, соответственно, делится на однофазный и двухфазный.

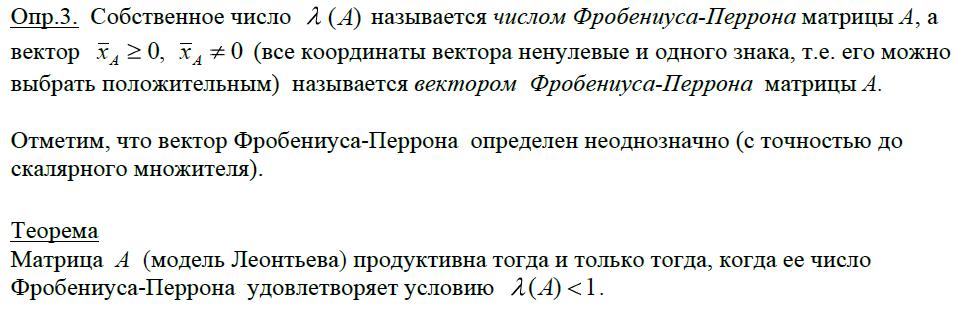
# 22. Описание статистической модели Леонтьева. Условие продуктивности.



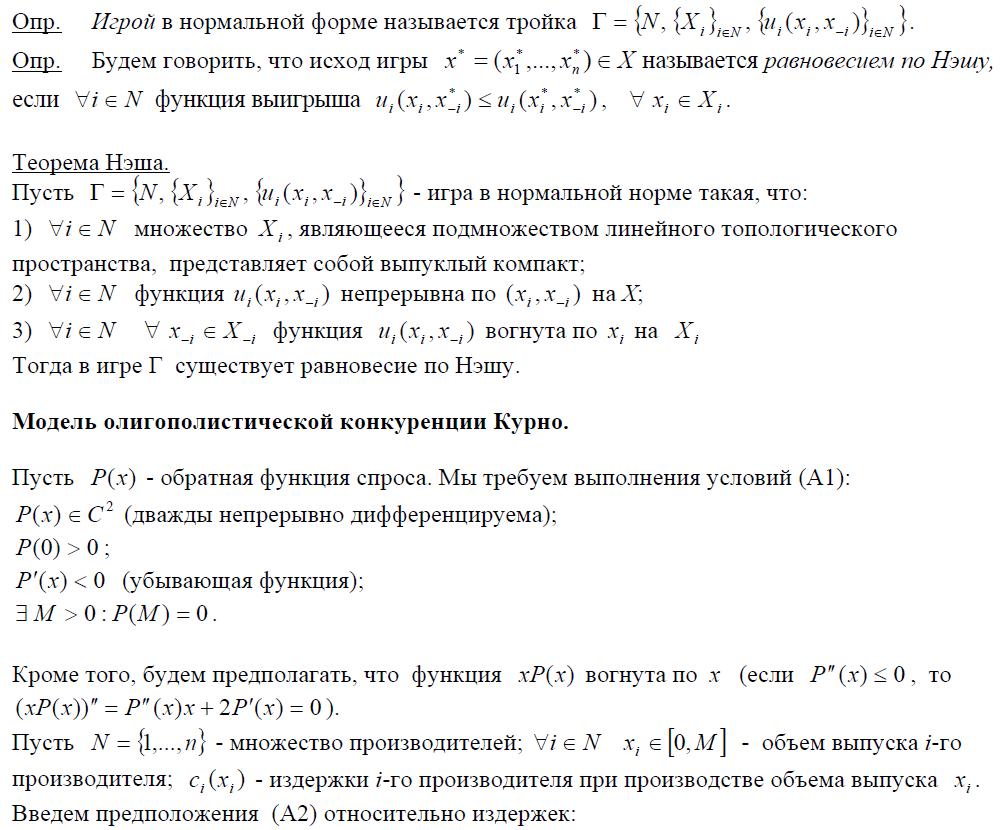


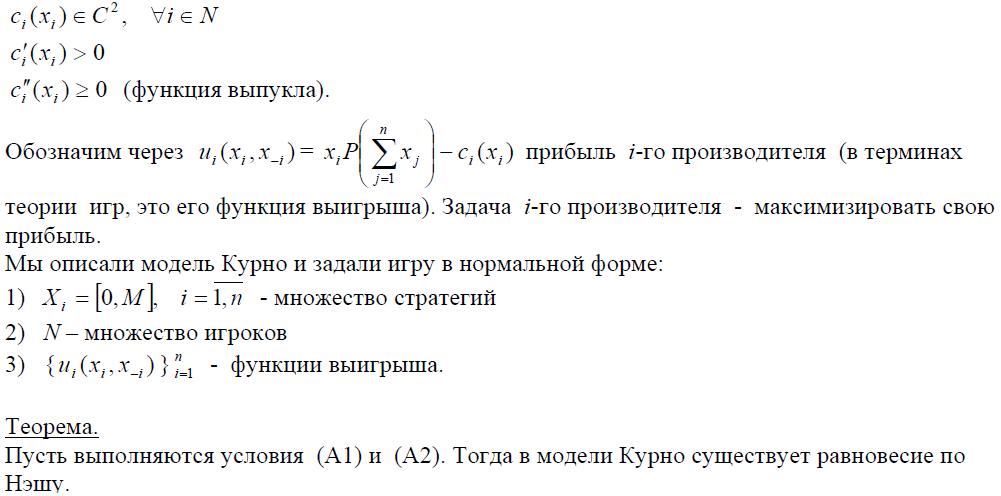
Условие продуктивности



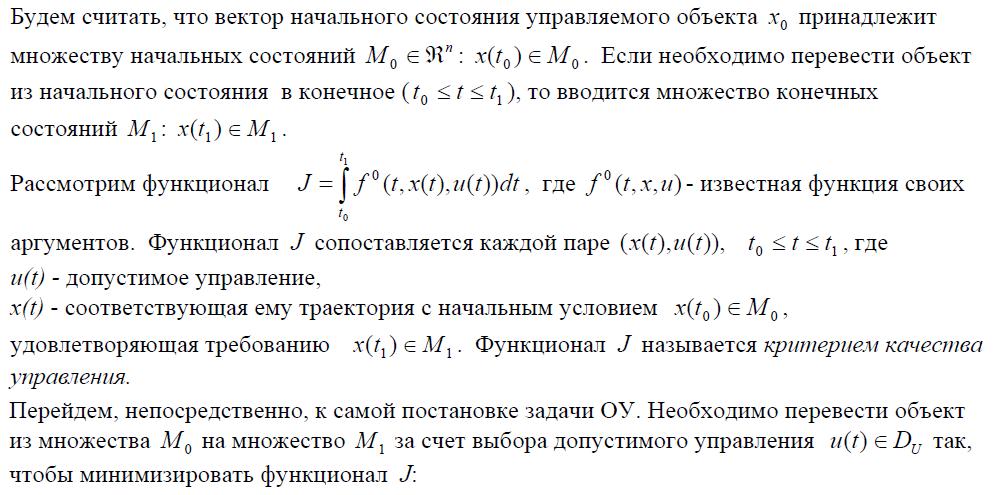


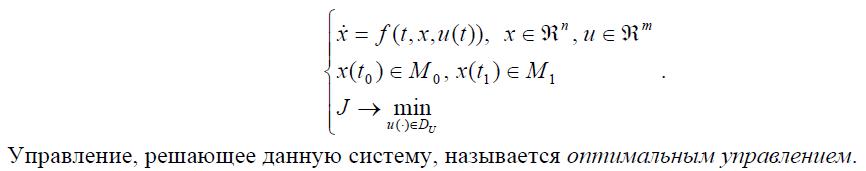
# 23. Модель Курно



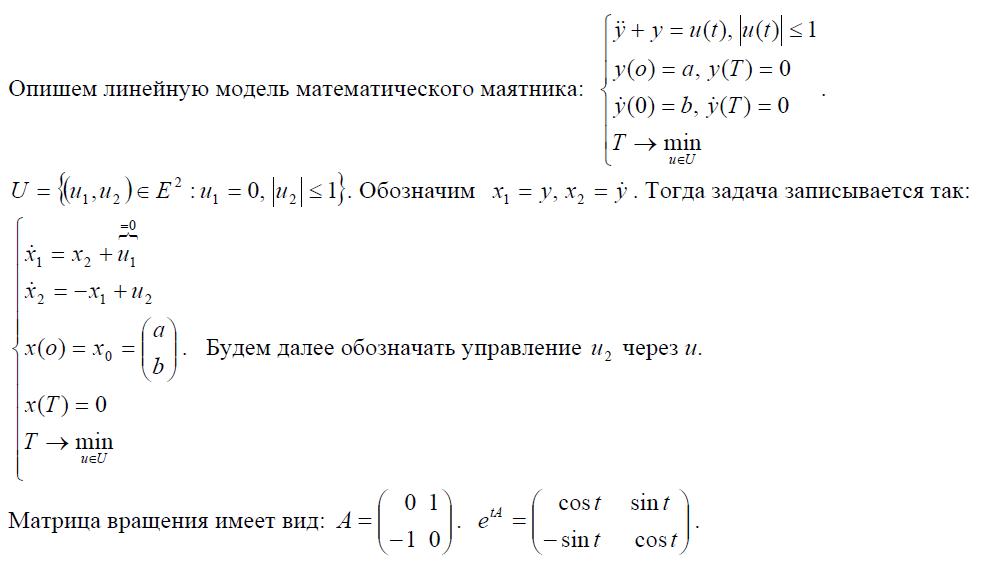


# 24. Постановка задачи оптимального управления. Понятие о задаче синтеза

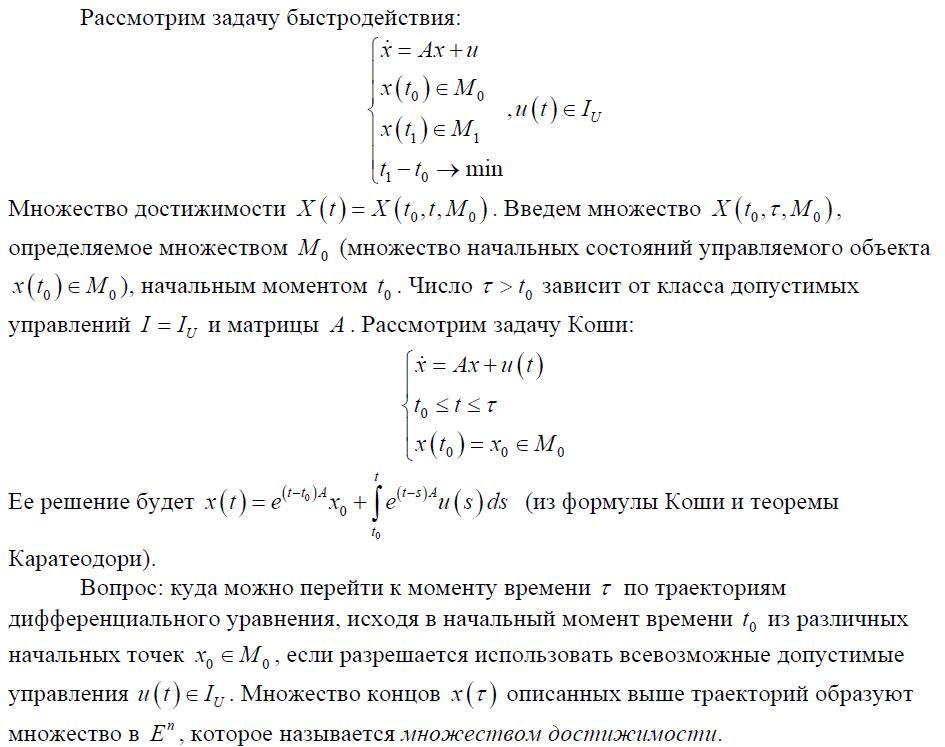


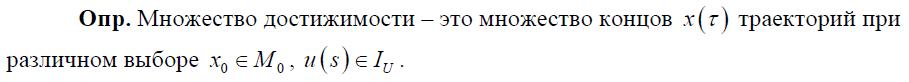


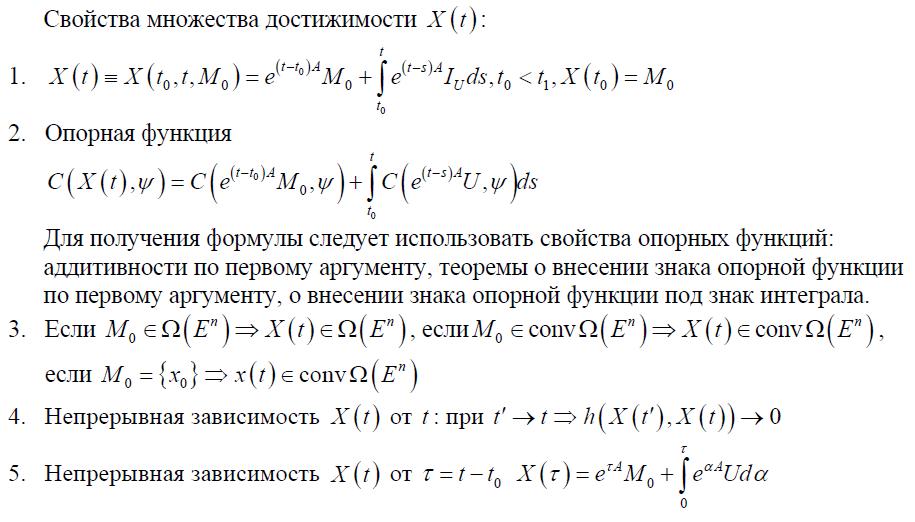
**Задача синтеза.**



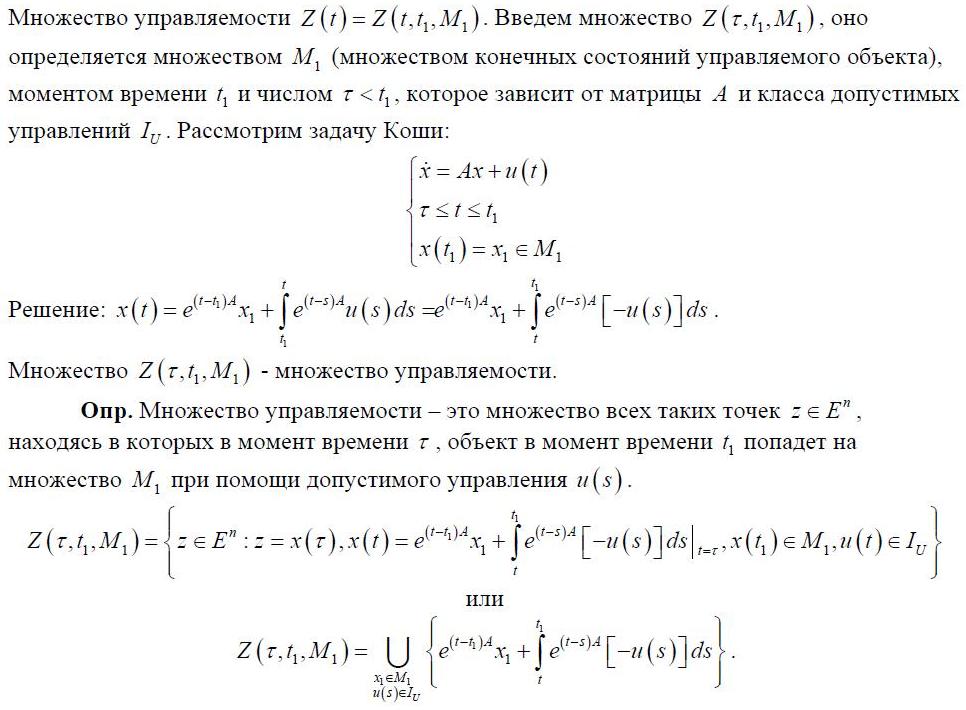
# 25. Множество достижимости линейной управляемой системы. Его опорная функция.

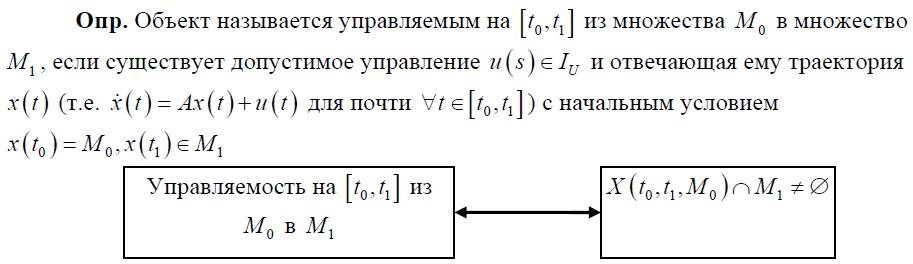




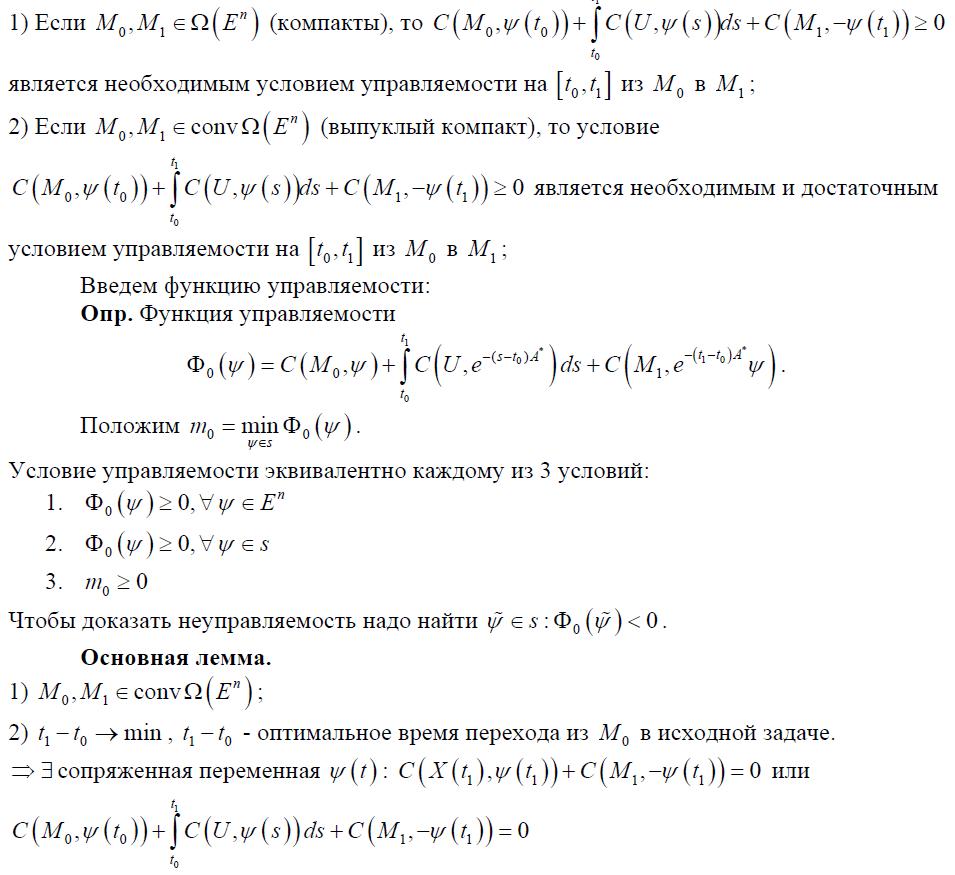


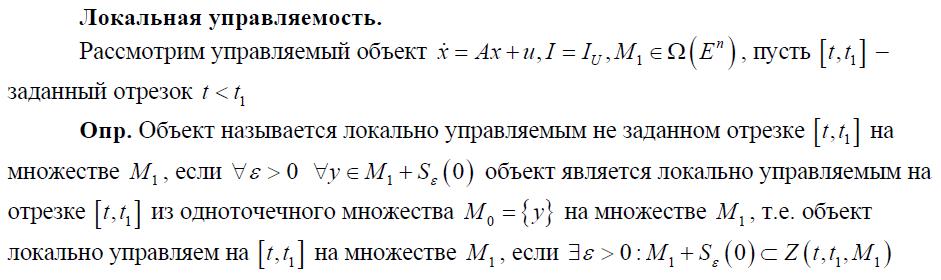
# 26. Управляемость и локальная управляемость линейных систем.

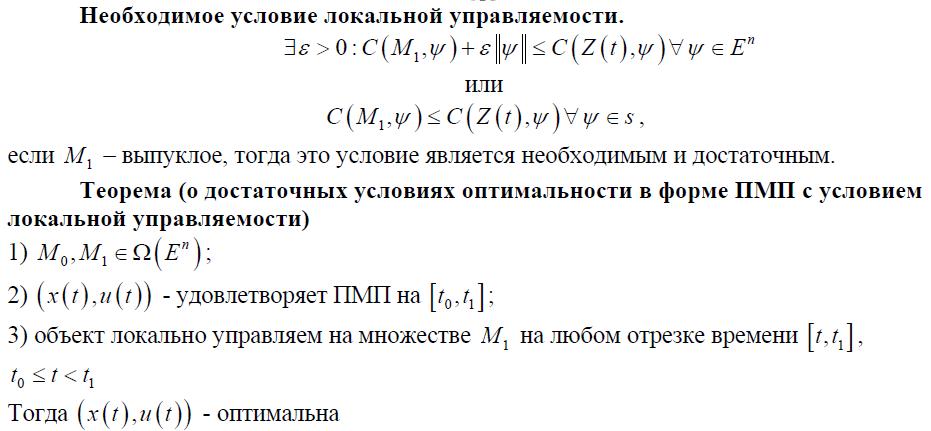




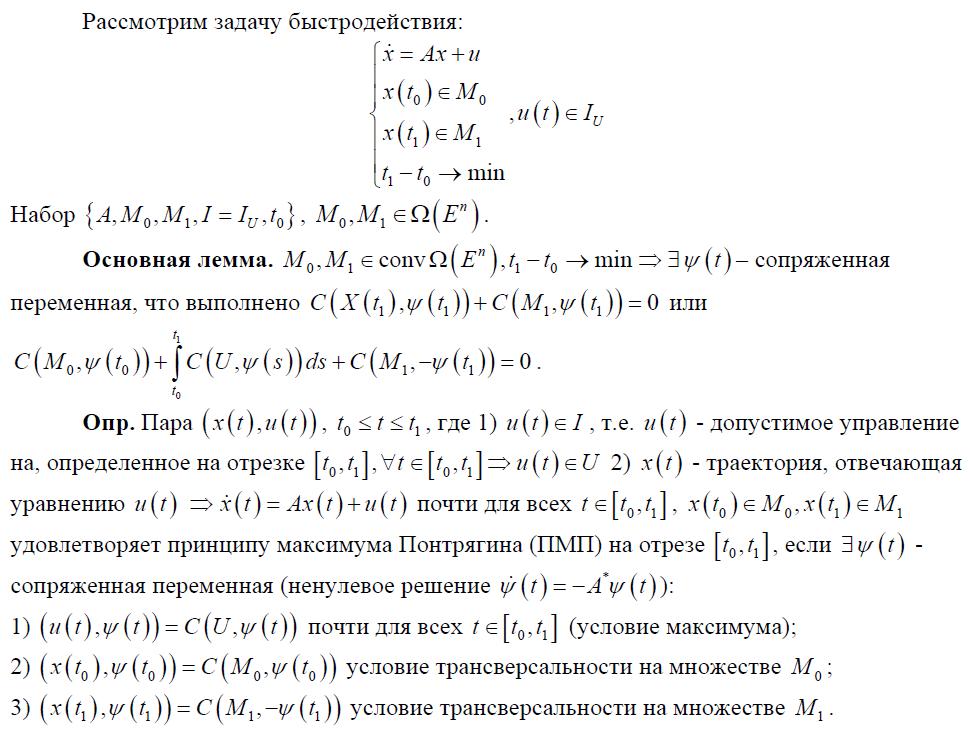
**Теорема (критерий управляемости).**

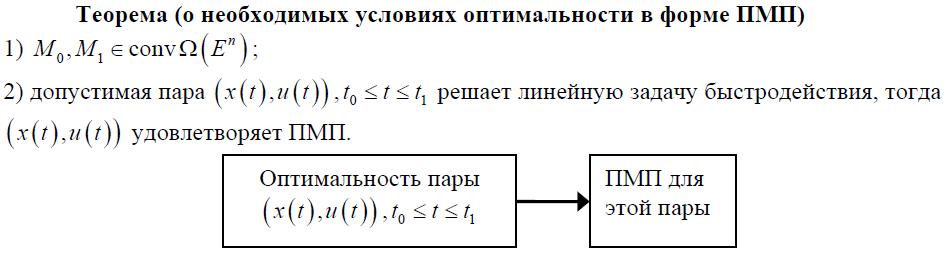




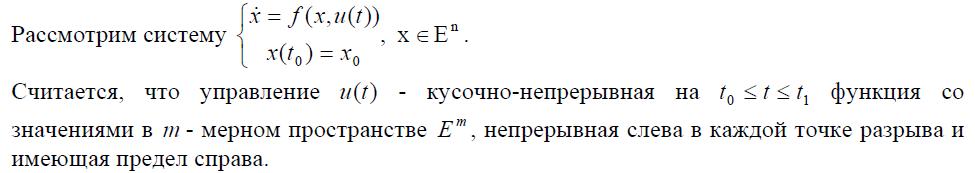


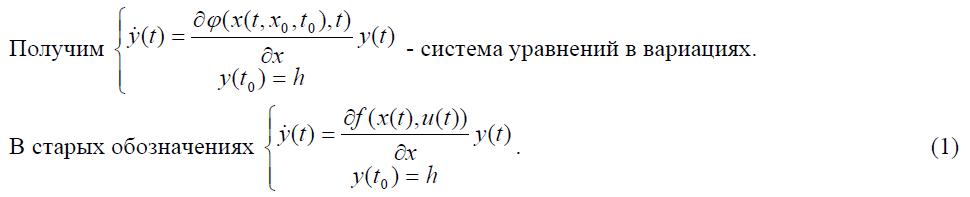
# 27. Принцип Максимума Понтрягина для линейной задачи быстродействия.

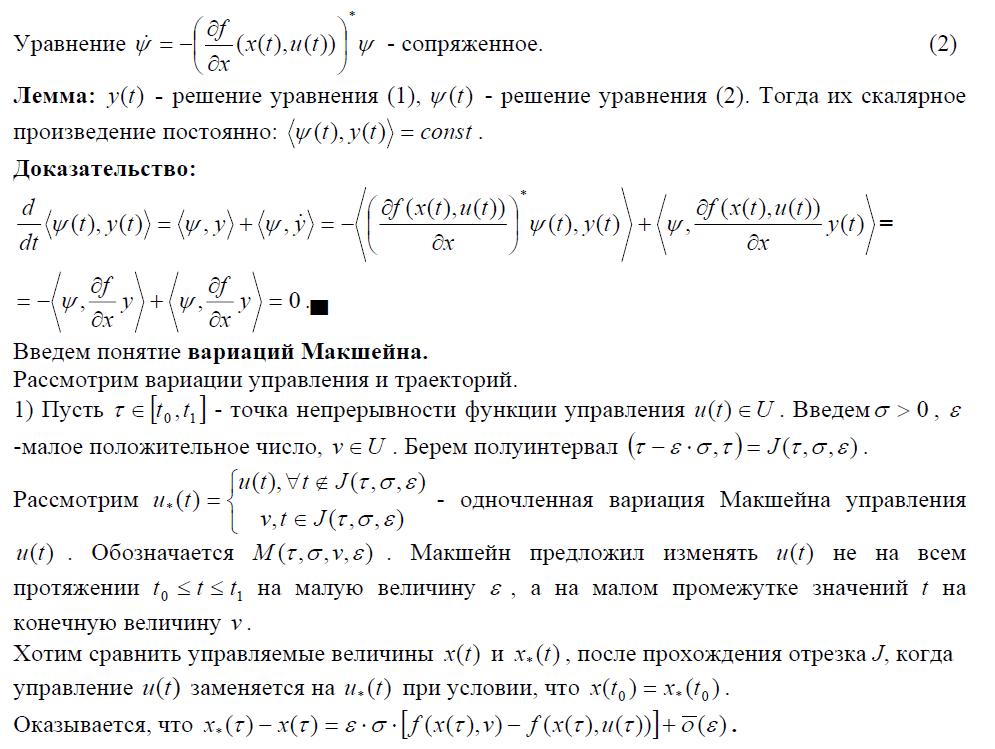


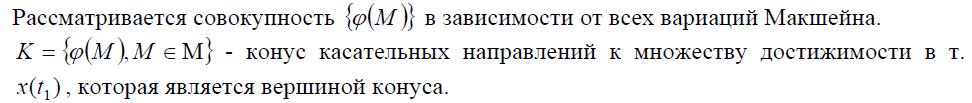


# 28. Уравнения в вариациях. Построение конуса касательных направлений к множеству достижимости.

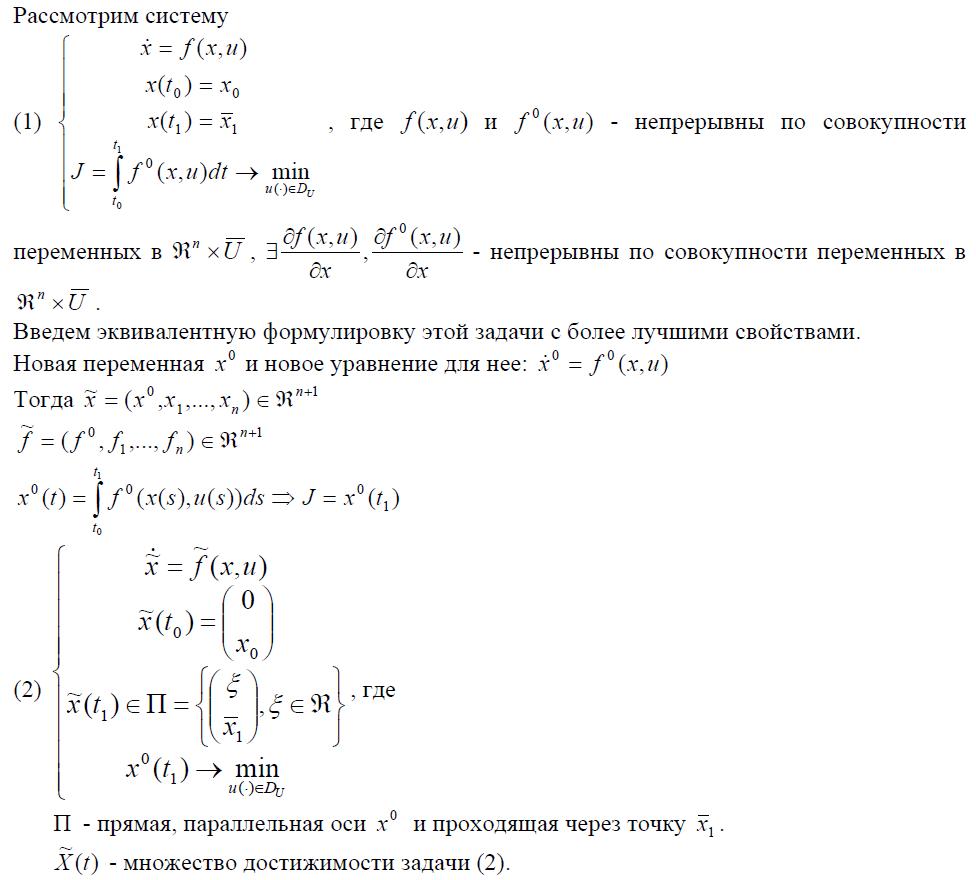


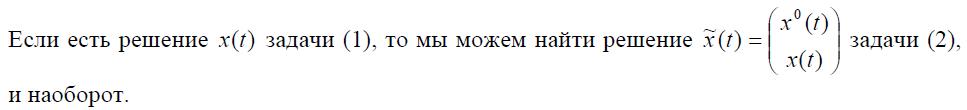


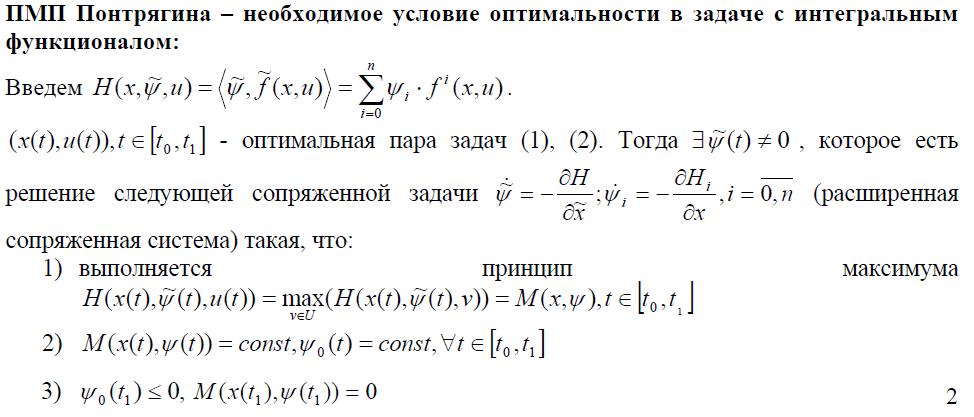




# 29. Принцип максимума Понтрягина для задачи оптимального управления с интегральным функционалом







# 30. Понятие о методе динамического программирования

