**2.Антиномии и их влияние на развитие МЛ.** На рубеже 19-20 в. были обнаружены антиномии, связанные с основными понятиями теории множеств. Анализ антиномии привёл к различным планам по её устранению.

Антиномии – ситуации, в к-рых противоречащие друг другу высказывания об одном и том же объекте имеют логически равноправное обоснование, т.е. нельзя чётко определить их истинность или ложность.

Антиномии:

а) Семантические (бытовые). Парадокс лжеца: «Я лгун».

б) Логические – сформулированы на строении языка. Парадокс Рассела:

A: X∉x(x∈A⇿x∉X)

A∈A→А∉A

А∉A→A∈A

Какой бы подход к исследованию парадоксов не выбрать, нужно сперва исследовать язык логики и математики, чтобы разобраться, какие символы могут быть используемы, как у них составляются теоремы, формулы, утверждения и т.д., что может доказано если исходить из тех или иных аксиом и правил вывода. Глубокие исследования различных математиков привели к завоеванию для МЛ положения независимой ветви математики.

**5. Кортеж (определение, прямое произведение, проекция).** Кортеж – неопределяемое понятие, обозначается “ <a, {b}, f(x,y)>”. Элементы кортежа имеют порядок и наз. компонетами. Кол-во компонентов назыв.длиной кортежа.

Кортеж длины 0 “Δ “, не существует кортежа длины 1. Два кортежа равны т.т.т.к.они имеют один.длину и компоненты равны между собой. <a1,..,an> = <b1,..,bn> ∀i ai=bi

Произведение: пусть ∀i Mi – множество, тогда можно задать новое множество M1 x M2 x … x Mn =  
Мi. |M| - мощность М. Мощность прямого произв. равна произв.мощностей этих множеств: |Mi|=|Mi|

Проекция: пусть α = <a1..an> - некоторый кортеж, тогда проекция на любую ось есть его компонента.

1. пр.i α = ai 1<=i<n
2. пр i, i2..in α=β=<ai1..ain> при этом 1<=i1<i2<i3<..<in<=n. Если условие нарушено, то проекция неосмыслена.
3. проекция на пустое множество пр. øα = λ

Для множ-ва кортежей одинаковой длины можно определить понятие проекции, при этом проекция такого мн-ва равна множ-ву проекций его элем.

**8. Функция.** Функция – это функциональное соответствие f=<F,X,Y>, F – функц. график, пр1F – обл.отправл.,пр.2F – обл. прибытия. f: X->Y.

Функция нигде не определена,если её график ø

Ф-ция определена: !f(a)

F(a)=b&&f({a}) = {b} где <a,b> ∈ F

Функция назыв. тождественной на мн-ве А, если диаг.нек-го мн-ва есть подмн-во графика.

Частные случаи ф-й:

1. F= ø – ф-ция нигде не определена
2. Im⇋<∆m,M,M> тожд. ф-ция
3. A ⊆пр1F ф-ция определена
4. отобр. X->Y – ф-ция задана на X и отобр. в Y. Если ф-я инъективна, то её инверсия функциональна.

f-1 ∘ f = пр2F

f ∘ f-1 = пр1F

Для любой ф-ции f существ. такая g, что ∀f∃g: g∘f=I пр2F

Обратная ф-ция: для f обратной найдётся ф-ия g такая, что

1. g∘f=I пр2F
2. g – инъекция
3. [пр1G = пр2F; пр2 G⊆пр1F] G⊆F-1

**11. Формальные (аксиоматические) теории. Их разрешимость.** Формальн.теория опр. алфавитом, ППФ, (нек. подмн-ва ППФ объявл. аксиомами), правилами вывода.

Вывод – последовательность формул типа ППФ, каждая из к-рых есть следствие из правил выбора или аксиома. Посл. форма из A1, A2.. An – теорема

Формальная теория наз. разрешимой, если сущ. универсальный метод, к-рый для ∀ формулы типа ППФ даёт ответ, является ли она теоремой или нет.

Теорема: (дедукции искл.вычислений). Если из нек-рой системы формул Г, к к-рой добавл. ф-ла A выводима ф-ла 𝔑, то из Г вывод. ф-ла Г ↦ A ↦ N.

Док-во: ∃ посл. формул 𝔑0, 𝔑1, … 𝔑i … 𝔑n ≡ 𝔑. Преобр. этот вывод формулы A-> N

𝔑0 -> а) аксиома N0 – базис индукции

б)эл-т ∈ Г

в) A -> вместо 𝔑0 G ф-л A-> 𝔑

N0, N0 -> (A -> N0), A-> N0

И для всех пред. ф-л получено A-> 𝔑0

𝔑i: а) аксиомаi

б) ∈Г

в) A

г) получена из ф-л по МР

𝔑j ≡ 𝔑1 -> 𝔑i

Правила вывода:

Modus ponendo ponens – Если A влечёт B и A, то B

Modus tollendo tolens – Если А влечёт B и не B, то не А

Modus tollendo ponens – Если А либо B и не А, то В

Modus ponendo tollens – Если неверно, что А и В, и А, то не В.

**14. Полнота исчислений высказываний.** ФТ назыв. относ. преобразованной a->a’, если

1) а) а – теорема

б) присоединение этой формулы в качестве аксиомы превращает науку ФТ в противоречивую относ. преобр.

Теория назыв. полной в абс. смысле, если произв. формула – теорема, присоед. ф-лы в кач-ве аксиомы превр. теорию в противореч. в абс.смысле, и если присоед. ф-лы в кач-ве аксиомы превр. теорию в противореч. в смысле поста. P1 полна во всех смыслах.

Док-во: 1) ʆ -> а)тавтология Мощность или кардинальное число, связная сумма, праймориал, примориал - математический символ 

б) не тавтология –> не теорема -> меняем ʆ по системам аксиом.

(P1) (3 аксиомы + ʆ), ʆ – не тавтология =>существ. интерпретация (b1..bn), где ʆ ≡0, ʆ (b1..bn), not ʆ подстановкой, (ζ – теор.) выводима из ʆ, подстановкой приводима к Л. ʆ – теорема либо противоречие. ʆ ->f (тавтология) -> ʆ ->f, P̅1 противоречива отн. преобраз. ʆ.

2) по М.Р, f – теорема, f – тавтология, М-Р: ∀ 𝔄i f-> 𝔄 – теорем., т.е. ∀ СФ – теорема P̅1 =>P̅1 противореч. в абс.смысле.

3) возьмём частн. случай A => ∀ пропозиц.переменная – теорема => P̅ - противореч. в смысле Поста. Тогда P̅1 полна во всех 3х смыслах.

**17. Интерпретация**. ⌟ имеются 2 ФТ: Т1 и Т2

Способ замены одного отнош. (предиката) на другое, одного класса зн-ий переменной на другой, при к-ром аксиомы Т1 будут истины для Т2 назыв. методом демонстрации модели или методом интерпретации. и Интерпретацией наз. пара <D, T>, где D – обл. интер-ии, из к-рой черпают значения переменных. Г – соотв. между любым предикатным символом и отнош. на D, функц. и функц. символом на D, переменным значением из D.

Изоморфизм интерпретаций: ⌟ интерпретация <D, Г> и <D’, Г’>, |D|=|D'|, кол-во констант, функц. и предикатных символов одинаково, эти символы можно упорядочить.

Изоморфизм для этих двух интерпретаций – взаимооднозначное отобр. g=<G,D,D’> и

1. ∀a∈D, g(a)=a’, a’∈D (a-const)
2. ∀Pjm (aj) ≡ P’jm(g(aj))
3. g(fin(ai)) = fi’n(g(aj))

**20. Приведенная и предваренная нормальные формы. Теоремы о них.** Приведенная формула – ф-ла, в к-рой отсутств. импликация, при этом отрицания стоят непоср. перед атомарн.формулами.

Теорема: любая ФИП представима в приведенной форме

Док-во: а) A -> B ≡ ¬ A v B

б) ¬ (A & B) ≡ ¬ A & ¬ B #

Предваренная нормальная форма

(матрица)

ʞ1x1 ʞ2x2.. ʞmxm 𝔞(x1..xm,y..)

Теорема: Для любой предикатной формулы существует равносильная ей формула в предваренной нормальной форме.

Д-во:

1) Если наша приведенная формула автоматически оказывается в ПНФ.

2) 1)-неверно. Применяется преобразование: {∀x 𝔑≡𝔑; ∃x𝔑≡𝔑 }, удалив часть кванторов, снова оказывается в ПНФ.

3) Либо не применима, либо не обратима в ПНФ: {∀xA(x1...)= ∀tA(t1...); ∃xA(x1...)= ∃tA(t1...)}

**23. Теорема Геделя о полноте.** Для того, чтобы формула ИП первого порядка была теоремой, Н и Д, чтобы она была логически общезначимой.

Не существует общего алгоритма, способа, метода, с помощью к-рого можно было установить: является ли произв. формула МП первого порядка теоремой или нет.

**26. Примитивно рекурсивные функции и предикаты.** Заданы начальные ф-ции:

z(x1..xn)=0;N(x)=x+1; δin(x1..xn)=xi

Способы получения новых ф-ций:

1. Подстановка n(x1..xn)=g(f1(x1..xn)..fk(x1..xn))
2. Рекурсия

f(y):

f(x1..xn, y):

1. μ – оператор

f(x1..xn)= μy(g(x1..xn,y)=0)

Примитивно-рекурсивной называется функция, которая может быть получена из начальных путём конечного числа подстановок и рекурсий.

Рекурсивной называется функция, которая может быть получена из начальных путём конечного числа подстановок, рекурсий и μ-операторов.

**29. Схема.** Пусть имеется нек-рый алфавит А, не содерж. символ ℐ. Тогда ∀ слово в Aj, начин. и заканч. символом ℐ, будет называться системой или ℐ-системой.

А – алфавит, ℐ - система, ℐPj – атом, Р – слово А

S\* ℐPj\*T – элемент системы

Активный элемент схемы для данного слова Р

1. Если он действует на Р
2. Если не предшествует ни один действ. элемент системы

Результат действия схемы на Р – результат действия на слово её активн. элемента.

**32. Нормальные алгоритмы (НА), применимость, замкнутость, композиция.**

Нормальный алгоритм определяется:

1. Алфавитом, в к-ром он работает (А)
2. 3-мя спец.симолами: α~→

которые не принадлежат β~→.

алфавиту ℐ~ck (возврат каретки и перевод строки)

1. Схемой, к-рая работает над словами (z)

σAσαβσzσ ~ A

Будем говорить, что алгоритм применим к слову Р, если за конечное число шагов он перерабатывает слово Р в слово Q и останавливается. Q – результат работы алгоритма A.

Любой сокращающий НА применим к любому слову и перерабатывает его за кол-во шагов, не превышающ. длину этого слова.

НА не применим ни к какому слову, если в словах нет символов, принадл. алфавиту алгоритма.

НА называется замкнутым, если в схеме имеется подстановка с пустой левой частью. Такой НА действует на любое слово. Естеств. завершенная у такого НА быть не может.

Композиция НАМ. A(Р) ∘ 𝔑(Q)= 𝔑(A(P))

Для любых НА A и 𝔑 существ. такой алгорифм 𝔐 над объед. алфавитами A и 𝔑, что 𝔐 (Р=𝔑(A(Р)), где Р – слово в объед. алфавите.

**35. Понятие о массовой алгоритмической проблеме.** Нормальная массовая проблема – построить нормальный алгоритм над алфавитом, содержащим все возможные символы входных наборов единичных задач, применимых к любому входному набору и перерабатывающей его в пустое слово т.т.т.к. это единственная задача разрешимая в положительном смысле. Теорема: Может быть построен НА 𝔐 в алфавите {a;b} такой, что не существует НА над алфавитом {a;b}, применимый к любому слову в этом алфавите и перераб. в пустое слово λ т.т.т.к. 𝔐 применим к этому слову. Проблема применимости НА неразрешима.

**38. Эрбрановский базис.** Мн-во атомов системы S для всех термов Эрбрановского универсума.

Пример: S={P(x) v Q(y); R(a,z) v T(w)}

B={P(a), Q(a), R(a,a), T(a)}

**41. Подстановка**. Композиция подстановок. Унификатор. Подстановка – конечная совокупн. ф-л подстановок.

η = {, , ..} xi – перем. системы, ti – любые термы

Композиция подстановок: пусть имеются 2 подстановки η и ξ, тогда тривиальн. формулы x/x вычёркив., из 2ой части вычёркив. формулы с совпад. значениями

η0={..…}

Унификатор – подстановка, примен. к двум формам, превр. их в одну и ту же форму.

НОУ – наиб. общий унификатор.

Подстановка назыв. основной, если все термы в этой подстановке явл. основными.

**44. Алгоритм поглощения.**

Р(а), Р(а) v Q(b)

P(x) υ{}

α ℓ υ

S={α, ¬ φ υ}={α, ¬Pi υ(i=1,k)}

W={¬Pi υ| i=1̅,k̅}

Алгоритм:

Шаг 0 k:=0, V0:={α}

Шаг 1 ☐∈Vk => Stop (+)

Шаг 2 Vk = ø => Stop (-)

Шаг 3 Vk+1 = {резольвента C1 и C2 | C1∈Vk, C2∈W}

Vk+1= ø -> Stop (-)

k:=k+1, goto 1