Билет 1. Понятие высказывания. Логические связки. Формулы логики высказываний.

Понятие высказывания.

Под высказыванием понимают всякое имеющее смысл повествовательное предложение, относительно которого можно утверждать истинно оно или ложно.

Логические связки.

Простые высказывания обозначаются пропозиционалными переменными, которые принимают истиностные значения И и Л. Построение из одного или нескольких высказываний называется логической операцией или логической связкой.

P	Q	P&Q	PVQ	P->Q	P~Q	P®Q
И	И	И	И	И	И	Л
И	Л	Л	И	Л	Л	И
Л	И	Л	И	И	Л	И
Л	Л	Л	Л	И	И	Л

Все символы называются пропозициональными связками.

Формулы логики высказываний.

Алфавит логики высказываний содержит следующие символы:

1. Высказывательные или пропозициональные переменные: $x_1, x_{2,\ldots}, x_{i,\ldots}$
2. Логические символы: ¬, V,&,->,~
3. (,)

Слово в алфавите логики высказываний называется формулой,если оно удовлетворяет следующему определению:

1. Любая высказывательная переменная – формула.
2. Если А и В – формулы, то $\overline{A}$, A&B, AVB,A→B, A~B – тоже формулы.
3. Формулами являются только те слова, для которых это следует из двух предыдущих условий.

Подформулой формулы А называется любое подслово А, само являющееся формулой.

Упорядоченный набор.

Упорядоченный набор высказывательных переменных $<x_1,\ldots x_n>$ назовём списком переменных формулы А,если все переменные этой формулы содержатся в этом наборе. В таком списке некоторые переменные могут быть фиктивными.

Конкретный набор.

Конкретный набор истиностных значений приписанных переменных назовём оценкой списка переменных или интерпритацией формулы А.

Билет 2. Равносильность формул логики высказываний. Основные равносильности.

Равносильность формул логики высказываний.

Пусть А и В формулы зависящие от одного списка переменных $<x_1,\ldots x_n>$, будем называть их равносильными, если на любой оценке этого списка они принимают одинаковые значения.

Основные равносильности.

1.A&A≡A; A→A≡И(закон сохранения); AVA≡A; A~A≡ И

2.Закон комутативности

A&B≡ $B\&A$; A~B≡B~A; AVB≡BVA; A→B≠B→A

3.Закон «лжи-истина»

A&И≡A; AVИ≡И

4.Закон противоречия

A&$\overline{A}$ ≡Л

5.Закон исключения третьего: AV$\overline{A}$ ≡И

6.Закон двойного отрицания: $\overline{\overline{A}} \equiv A$

7.Закон дистрибутивности:

A&(BVC)≡(A&B)V(A&C)

AV(B&C)≡(AVB)&(AVC)

A~(B~C) ≡(A~B)~C

A→(B→C) ≡(A→B)→C

8.Закон де Моргана: $\overline{A\&B} \equiv \overline{A} V \overline{B}$; $\overline{AVB} \equiv \overline{A}\&\overline{B}$

9.Закон поглощения: A&(AVB) ≡A; AV(A&B) ≡A

10. Формулы расщепления: A≡ (A&B)V(A&$\overline{B}$); A≡ (AVB)&(AV$\overline{B}$)

11. A→B≡ $\overline{A}$VB≡ $\overline{A\&\overline{B}}$; AVB≡ $\overline{A} \rightarrow B \equiv \overline{\overline{A}\&\overline{B}}$; A&B≡ $\overline{A \rightarrow \overline{B}} \equiv \overline{A} V \overline{B}$;

A~B≡(A→B)&(B→A)≡(A&B)V($\overline{A}$&$\overline{B}$)≡(BV$\overline{A}$)&(AV$\overline{B}$)

Билет 3 Тождественно-истинные формулы логики высказываний. Важнейшие тавтологии. Правильные рассуждения. Утверждение о правильности рассуждения по схеме $(P_1 \dots P_n) \Rightarrow Q$

Тождественно-истинные формулы логики высказываний

Формула А- тавтология (или тождественно истинная формула или общезначимая) если на всех оценках списка своих переменных оно принимает значение «И».

Формула А-выполнимая, если хотя бы в одной своей интерпритации она принимает значение «И»

Формула А-тождественно ложная или противоречивая если на всех оценках списка своих переменных она принимает значение «Л».

С точки зрения мат.логики тавтологии по сути являются логическими законами, т.к в любой подстановке истинных высказываний получаются истинные высказывания.

Наиболее важные тавтологии:

1. $A \vee \overline{A} = И$
2. $A \to A = И$
3. $(A \,\&\, B) \to A \qquad (A \,\&\, B) \to B$
4. $A \to (A \vee B) \qquad B \to (A \vee B)$
5. $A \to (B \to A)$
6. $A \to (B \to (A \,\&\, B))$
7. $(A \to B) \to ((B \to C) \to (A \to C))$ цепное рассуждение
8. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
9. $(\overline{A} \to \overline{B}) \to ((\overline{A} \to \overline{B}) \to A$
10. $((A \to B) \to A) \to A$ закон Мерса

Правильные рассуждения

При доказательстве утверждений математических рассуждений используют рассуждения, которые на языке логики можно выразить формулами. Рассуждение называется правильным, если из коньюнкции посылок следует заключение. Таким образом правильность рассуждений можно установить, составив соответствующую ему формулу логики высказываний и доказать, что она является тавтологией.

Утверждение о правильности рассуждения по схеме $(P_1 \dots P_n) \Rightarrow Q$

Пусть $P_1 \dots P_n \Rightarrow Q$ посылки, Q- заключение.

Утверждение.

Рассуждение по схеме $P_1 \dots P_n \Rightarrow Q$ правильны тогда и только тогда, когда $(P_1 \,\&\, \dots \,\&\, P_n) \to Q$ тавтология.

Доказательство

1. $\to$

Пусть рассуждение по этой схеме является правильным, допустим $I((P_1 \,\&\, \dots \,\&\, P_n) \to Q) = И$ по определению импликаций. Если же $I(P_1 \,\&\, \dots \,\&\, P_n) = Л$, то $I(P_1 \,\&\, \dots \,\&\, P_n \to \mathrm{Q}) = И$ при любых Q.

Следовательно формула $(P_1 \,\&\, \dots \,\&\, P_n) \to Q$ тавтология.

2. $\leftarrow$

Пусть $(P_1 \,\&\, \dots \,\&\, P_n) \to Q$ - тавтология. Допустим $I((P_1 \,\&\, \dots \,\&\, P_n) \to Q) = И$, тогда $I(Q) = И$, иначе $P_1 \,\&\, \dots \,\&\, P_n \to Q$ - не тавтология. Таким образом формула Q есть логическое следствие формулы $P_1 \,\&\, \dots \,\&\, P_n$, т.е рассуждение по этой схеме $(P_1 \,\&\, \dots \,\&\, P_n) \Rightarrow Q$-правильное.

Билет 4 Проблема разрешимости в логике высказываний и методы ее решения.

Вопрос к какому классу формул (тавтология, выполнимая или тождественно-логичная формула) относится текущая формула А называют проблемой разрешимости, которая элементарно решается с помощью таблицы истинности. Однако для формул таблицы громоздкие и их исчисление затруднительно.

Другой способ основан на приведение формулы А к КНФ или ДНФ использовании специального алгоритма, который позволяет определить являются ли данная формула тождественно-истинной или нет. Одновременно решается проблема разрешимости

$$\left.\begin{array}{l} f(x_1,\dots,x_n) = \cup\, x_1^{б1} \,\&\, \dots \,\&\, x_n^{б2} \\ f(б_1,\dots,б_n) = 1 \end{array}\right\} \text{совершенная ДНФ}$$

$f(x_1,...,x_n) = \&(\overline{x_1}^{б1} \vee ... \vee \overline{x_n}^{бn})$
$f(б_1,...,б_n) = 0$
совершенная КНФ

Правила перехода от СКНФ к СДНФ

СКНФ f= СДНФ $\overline{f}$

Вышеназванный алгоритм таков: сначала рассматривается формула А

Если А≡И $\Rightarrow$ тогда задача решена

Если $\overline{A} \equiv И \Rightarrow A \equiv Л \Rightarrow$ решена

Если $\overline{A} \neq И \Rightarrow$ А-выполнимая

Установление тождественной истинности формулы А основано на следующих теоремах:

1. Для того чтобы элементарная дизъюнкция была тождественно истинной необходимо и достаточно, чтобы в ней содержались переменная и ее отрицание:

$x_i \vee \overline{x_i} \equiv И$

2 .Для того чтобы элементарная конъюнкция была тождественно ложной необходимо и достаточно чтобы в ней содержались переменная и ее отрицание:

$x_i \& \overline{x_i} \equiv И$

3. Для того чтобы формула высказывания А была истинной необходимо и достаточно чтобы любая элементарная дизъюнкция входящая в КНФ А содержала переменную и ее отрицание.
4. Для того чтобы формула высказывания А была тождественно ложной необходимо и достаточно чтобы любая элементарная конъюнкция, входящая в ДНФ А содержала переменную и ее отрицание.

Билет 5. Определение и виды формальных теорий.

\\Обозначения : ≤ - «входит или равно», < - входит (про множества).Т.к. знаков не нашел таких в ворде

Рассмотренные ранее таблицы истинности позволяют ответить на многие вопросы, касающиеся формул логики высказываний : например о тождественной истинности формулы или равносильности двух формул. Однако для исследования более сложных вопросов приходится использовать более сложный метод : метод формальных аксиоматических теорий.

Формальная теория (исчисление) Т считается заданной, если определены след компоненты:

1. А – алфавит
2. F≤A* - мн-во слов и букв алфавита А – формулы теории
3. Подмножество В формул B<F – аксиомы
4. Кроме того между формулами задается мн-во отношений R, кот. Называются правилами вывода, они используются для изложения всех теорем в данной теории.

Алфавит А может быть как конечным, так и бесконечным. Чаще всего для образования символов используют конечное множество букв, к которым приписываются, если нужно, натуральные числа в кач-ве индексов.

Мн-во формул обычно задается индуктивным опред-ем, как правило оно конечно.

В совокупности А и F определяют язык (или сигнатуру) форм. Теории.

Мн-во аксиом может быть конечным или бесконечным. Если мн-во бесконечно, то оно задается, как правило, с помощью конечного мн-ва схем аксиом и правил порождения конкретных аксиом из схем аксиом. Мн-во правил вывода обычно конечно.

Пусть F1,…,Fn,G – формулы теории Т. Если сущ. Такое правило вывода r, что (F1,…,Fn,G) ∈ r, т.е. формулы F1,…Fn,G находятся в отношении r, то G непосредственно (в одно действие) выводима из F1,…,Fn по правилу вывода r. Обычно факт записывается:

[(F1,…,Fn)/G]r, Теория назыв формально непротиворечивой

где F1,…,Fn – посылки правила r; G – следствие (заключение) правила r.

Выводом ф-лы G из ф-лы F1,..,Fn называется такая послед-ть формул E1,..,Ек, что Ек=G, а люб. Ei(i<k) – либо аксиома, либо одно из исходных формул, либо непосредственно выводима из ранее полученных формул E по одному из правил вывода.

Если в теории Т сущ. Вывод формулы G из формул F1,..,Fn то говорят, что G – выводима из F1,..,Fn. Этот факт обозначают так : F1,..Fn |---$_T$ G, где ф-лы F1,..Fn назыв гипотезами или посылками вывода.

Д-вом ф-лы G в теории Т назыв вывод G из пустого мн-ва формул, т.е. без гипотез, используя только аксиомы.

Формула G для кот сущ док-во называется формулой доказуемой в теории Т или же – теоремой теории Т.

|---$_T$ G – G теорема теории Т, G доказуема в теории Т

Очевидно, что при соединении формул гипотезой теории не нарушается : F1,..Fn |--- G , то

F1,..,Fn,Fn+1 |--- G

порядок гипотез в списке несущественен.
Формальная теория называется полной, если для любой ф-лы А имеем: |---А или же
|---Ā
В полной теории любая правильно построенная формула должна быть теоремой, т.е. должна быть доказуема утверждением.
Форм. Теория назыв. Симантически не противоречивой если ни одна ее теорема не является противоречием.
Форм. Теория назыв формально непротиворечивой если в ней не явл-ся выводимыми одновременно ф-лы F и $\bar{F}$. Можно доказать что теория формально непротиворечива, тогда и только тогда, когда она семантически непротиворечива.
Формальная теорема (ФТ) назыв разрешимой если существует алгоритм кот для любой формулы теории определяет, явл-ся ли эта формула теоремой теории.
Замечание: при изучении ФТ имеем дело с двумя типами высказываний:

1. Высказ-ий самой теории теоремы, кот рассматр как формальные объекты, определенные ранее.
2. Выск-я о теории – о св-вах ее теорем, док-в и т.д. , кот формулируются на языке метаязыке внешенем **или какое то другое слово, хуй пойми, почерк не разобрал**\\ по отношению к теории и назыв метатеоремами. (Лучше в лекциях посмотрите ☺)

Различия между теоремами и метатеоремами будем указывать явно не всегда, но его всегда надо иметь в виду. \\ **мы чо в битве экстрасенсов участвуем??**\\
Например : из F1…Fn построили вывод G, тогда F1,..F2 |---G – метатеорема – ее можно присоединить к уже имеющимся правилам вывода и исп-ть как дополнительное правило вывода.

Билет 6. Язык, системы аксиом и основные правила вывода исчисления высказываний.

Исчисление высказываний – аксиоматическая логическая система, кот описывает тождественно истинные логические схемы, а ее интерпретация это алгебра высказываний.
1.Алфавит исчисления высказываний состоит из пропозициональных переменных (буквы латинского алфавита с индексами или без); знаков логических связок (&,¬, ∨ ,→); символов (,); операция следования |--- .
2. Формула исчисления высказываний – назыв слово алфавита исч выск-й удов условиям :
-пропозициональная переменная явл ф-лой (элементарная атомарная)
-если А и В – формулы, то Ā, А∨В, А→В тоже ф-лы
- других формул нет
Подформулой А формулы В исчисления высказываний назыв подслово А слова В само являющееся формулой
3.Аксиомы.
Системв аксиом I (Клини, 1952г):

I.1 А→(В→А)
I.2 (А→(В→С))→((А→В)→(А→С))
I.3 (А&В)→А
I.4 (А&В)→В
I.5 А→(В→(А&В))
I.6 А→(А∨В)
I.7 В→(А∨В)
I.8 (А→С) →((В→С)→((А∨В)→С))
I.9 (А→В)→((А→$\bar{B}$)→Ā)
I.10 (из двойного отрицания А следует А)

Другая система исп-ет только 2 логич связки : отрицание и импликацию, при этом & и ∨ рассм не как связки исч выска-й, а кк сокращения, употребление кот удобно, но не обязательно (одновременно & и ∨ исключаются из алфавита высказываний)
В этом случае А∨В≡Ā→В; А&В≡отрицание(А→В)
В рез-те система аксиом становится компактнее
Система II.
II.1≡I.1
II.2≡I.2
II.3≡(Ā→$\bar{B}$)→((Ā→В)→А)
Приведенные системы аксиом равносильны в том смысле, что порождают одно и то же множество формул. Д-во такого утв-я состоит из д-ва выводимости всех аксиом системы II из системы I и наоборот (с учетом замечания для & и ∨).

Система II компактнее,след-но и более компактны док-ва ее св-в, но в сист I короче вывод различных формул.
Замечание : возможны и другие сист аксиом равносильные первым двум.
4.Правила вывода
1)Правило подстановки : если α – выводимая формула, содержащая букву A(т.е. α(A)),то выводима и формула α(β), получающаяся из α заменой всех вхождений на произвольную формулу β : α(A)/α(β)
2)Правило заключения(Modus Ponens – MP):
Если α и α→β – выводимые формулы, то β тоже выводимая : (α,α→β)/β
Замечание :
В этом описании исчисления высказываний аксиомы явл-ся формулами исчисления, а в правилах вывода исп-ся метаформулы (схемы ф-л)
Схема фор-л α→β обозначает мн-во всех тех формул исчисления кот получаются если ее метапеременные заменить формулами исчисления :
α зам на A, β на A&B, то из α→β получим A→(A&B).

Билет 7. Производные правила вывода в ИВ: выводимость A→ A, правило введение импликации, транзитивность выводимости.

1.Выводимость $\vdash$ **A→ A, при ∀ A.(в теории L). Док-во:**

1) подставим в(Ⅱ2): B | a→a, C | a получаем (A→((A→A)→A))→((A→(A→A))→(A→A)).

2)Подставим в(Ⅱ1): B | a→a: A→((A→A)→A)

3) из (1) и (2) по MP получим выводимость (A→(A→A))→(A→A)

4)в (Ⅱ1) подставим B | a: A→A(A→A)

5) из (4) и(3) по правилу MP получим выводимость A→A.

2. Т. в теории L **: A | - B→A. Док-во:**
1) A – гипотеза по условию .

2) из (1) и аксиомы (Ⅱ.1) по правилу MP получим выводимость B→A при гипотенузе A.

Полученную выводимость можно вместе с правилом подстановки рассматривать как новое произв. Правило вывода – правило введения импликации**:α/β→α для ∀ β.**
3. Транзитивность выводимости*:* Если изγ вытекает α(γ | -α) и из α вытекает β(α | -β), то из γвытекает β(γ | -β). Это свойство непосредственно вытекает из определения выводимости.

Билет 8. Производные правила вывода в ИВ: теорема дедукции (без доказательства), правило силлогизма, правило введения отрицания.

1.Теорема дедукции**:** Если Γ,α | -β, то Γ | -α→β, и наоборот**.**

2.следствие из т. Дедукции – правило силлогизма**:** A→B, B→C, | - A→C(в теории L).

Док-во: построим вывод из этих гипотез:

1) A→B- гипотеза по условию,
2) B→C – гипотеза по условию
3) A- гипотеза
4) Из (3) и (1) по правилу MP получим: A, A→B | -B
5) Применим MP для (4) и (2): B, B→C | -C
6) Из (1),(3) и (5) имеем :A→B, B→C, A | -C
7) Из (6) по теореме дедукции: A→B, B→C | -A→C.

3. Правило введения отрицания (метод док-ва от противного)**:** Если Γ,A | -B и Γ,A | -$\overline{B}$, то Γ|- $\overline{A}$

1) по т. Дедукции, если Γ,A | -B и Γ,A | -$\overline{B}$, то Γ | -A→B и Γ | →A→$\overline{B}$

2)из (1) и (Ⅰ.9) двойным применением МП получим Γ | -$\overline{A}$.

Билет 9 Лемма для теоремы об общезначимых формулах исчисления высказываний. (Обозначение |--(L внизу) я не нашёл, так что не путать с минусом!)

Лемма: Из гипотез $A_1',...,A_n' \vdash_L A'$

Док-во:
ММИ по структуре формулы А

1. Переменная формула. Пусть А=а, тогда $a \mid\!-_L a, \overline{a} \mid\!-_L \overline{a}$

2. Отрицание: пусть $A = \overline{B}$. Возможны 2 случая: 1. пусть I(B)=И, тогда I(A)=Л и A'= $\overline{A} = \overline{\overline{B}}$. По индукционному предположению $A_1', \ldots, A_n' \mid\!-_L B$. Но в исчислении высказываний есть $\mid\!-_L B \to \overline{\overline{B}}$ по теореме 3а => $A_1', \ldots, A_n' \mid\!-_L \overline{\overline{B}} = A'$ 2. пусть И(B)=Л, тогда I(A)=И и A'= $A = \overline{B}$ по индукционному допущению из гипотез $A_1', \ldots, A_n' \mid\!- A' = \overline{B}$

3. Пусть А=B→C по индукционному предположению из гипотез $A_1', \ldots, A_n' \mid\!- B'$ и $A_1', \ldots, A_n' \mid\!- C'$

1. Пусть I(B)=Л, тогда для ∀ C I(A)=И и $B' = \overline{B}$, а A'=A, но $A_1', \ldots, A_n' \mid\!-_L \overline{B}$ (по индукционному допущению) и по теореме 3Б $\mid\!-_L \overline{B} \to (B \to C)$. => из гипотез $A_1', \ldots, A_n' \mid\!-_L A' = B \to C$ по правилу MP

2. Пусть I(B)=И, I(C)=И, тогда I(A)=И, C'=C, A'=A=B→C. Имеем $A_1', \ldots, A_n' \mid\!-_L C$ (по индукционному предположению) и $\mid\!-_L C \to (B \to C)$ по аксиоме 1 с подстановкой С. => по правилу MP
$A_1', \ldots, A_n' \mid\!-_L A' = B \to C$

3. Пусть I(B)=И, I(C)=Л. Тогда $A' = \overline{A} = \overline{B \to C}$, B'=B, C'= $\overline{C}$. Имеем из гипотез: $A_1', \ldots, A_n' \mid\!-_L B$ и $A_1', \ldots, A_n' \mid\!-_L \overline{C}$ (по индукционному допущению) и $\mid\!-_L B \to (\overline{C} \to \overline{B \to C})$ (по теореме 3д) отсюда 2 раза применяя правило заключения(MP) получим $A_1', \ldots, A_n' \mid\!-_L A' = \overline{B \to C}$.

Билет 10 Теорема об общезначимых формулах в исчислении высказываний

Теоремами исчисления высказываний являются тождетвенно истинные формулы и только они:
$\mid\!-_L A \Leftrightarrow A$-тавтология.

Док-во

1.(->) Аксиомы любой системы являются тавтологией. Правило заключения сохраняет тавтологичность => теоремы исчисления высказываний это тавтология.

2.(<-) Пусть формула А-тавтология, тогда $A_1', \ldots, A_n' \mid\!-_L A$ в любой интерапритации (по лемме), таким образом имеется 2^n различных выводимостей: $A_1', \ldots, A_n' \mid\!-_L A$ среди них есть 2 различающиеся по значению A'n принадлежит либо a_n либо $\overline{a_n}$: $A_1', \ldots, A_{n-1}', a_n \mid\!-_L A$ и $A_1', \ldots, A_{n-1}', \overline{a_n} \mid\!-_L A$ по теореме дедукции отсюда имеем $A_1', \ldots, A_{n-1}' \mid\!-_L a_n \to A$ и $A_1', \ldots, A_{n-1}' \mid\!-_L \overline{a_n} \to A$ по теореме 3е выполняется следующее $\mid\!-_L (a_n \to A) \to ((\overline{a_n} \to A) \to A)$ применяем 2 раза правило MP, получаем $A_1', \ldots, A_{n-1}' \mid\!-_L A$. Повторить весь процесс ещё n-1 раз докажем выводимость $\mid\!-_L A$ => А выводимо из не нулевого множества гипотез => А теорема.

Билет 11. Метод резолюций в исчислении высказываний.

В основе метода резолюций лежат следующие рассуждения:
для доказательства выводимости $\{A_1, \ldots, A_n\} \vdash A$ необходимо доказать, что $A_1 \& \ldots \& A_n \to A$ - тавтология.

Опр. Пусть $A_1 = A_1' \vee B$, $A_2 = A_2' \vee B$, $A_1' \vee A_2'$ - резольвента дизъюнктов A_1 и A_2 по переменной B и обозначается $res_B(A_1, A_2)$

Если переменная, по которой берется резольвента не важна, то $res(A_1, A_2)$. Если дизъюнкты не содержат противоположных переменных, то резольвент у них не существует $res(A, \overline{A}) = 0 \vee 0 \equiv 0$.

Опр. Пусть Г={$A_1, \ldots, A_n$}-множество дизъюнктов. $B_1, B_2, \ldots, B_n$ - резолютивный вывод из Г , если ∀ B_i(i=1,..n) выполняется :

1. $B_i \in \Gamma$
2. Существует $j, k < i$: $B_i = res(B_j, B_k)$

Теорема о полноте метода резолюций.

Множество дизъюнктов Г противоречиво только в том случае, когда существует резолютивный вывод из

Г заканчивается 0. **Без док-ва**.

Применим эту теорему для проверки выводимости формулы А из множества формул Г=$\{A_1, \dots, A_n\}$.

Докажем, что условие $\{A_1, \dots, A_n\} \vdash A$ равносильно условию $\{A_1, \dots, A_n, \overline{A}\} \vdash$, т.е. данная система противоречива.

Док-во.

Если $\{A_1, \dots, A_n\} \vdash A$, то $A_1\&, \dots, \&A_n \to A \equiv 1$, тогда $\overline{\overline{A_1\&, \dots, \&A_n} \vee A} \equiv 0, A_1\&, \dots, \&A_n \vee \overline{A} \equiv 0$, то есть $\Gamma_1 = \{A_1, \dots, A_n, \overline{A}\} \vdash$ Пусть $A_1\&, \dots, \&A_n\&\overline{A}$ обозначим через В приведением В к КНФ:

В=$C_1\&, \dots, \&C_m$, тогда задача проверки выводимости: $\{A_1, \dots, A_n\} \vdash A$ свелась к проверке противоречивости множества дизъюнктов $\Gamma_1' = \{C_1\&, \dots, \&C_m\}$, а последнее согласно теории о полноте метода резолюций равносильно существованию резолютивного вывода 0 из Γ_1'.

На практике в общем случае этот метод неэффективен, т.к. число переборов резольвент может быть очень большим, оно экспоненциально зависит от числа дизъюнктов и содержащихся в них переменных. Однако метод успешно применяется к хорновским дизъюнктам.

Дизъюнкт называют хорновским, если он содержит не более одной переменной степени 1(без $\overline{\ }$)

В общем случае хорновский дизъюнкт имеет вид

$\overline{A_1} \vee, \dots, \vee \overline{A_n}$ - негативный дизъюнкт.

$\overline{A_1} \vee, \dots, \vee \overline{A_n} \vee$B – точный дизъюнкт.

Дизъюнкт А- унитарный позитивный, $\overline{A}$ - унитарный негативный.

Если Г- множество хорновских дизъюнктов, то его проверка на противоречивость происходит следующим образом: в Г выбирается унитарный дизъюнкт В(или $\overline{B}$) и дизъюнкт С, который содержит $\overline{B}$ (или В) Далее Γ заменяется на $\Gamma \backslash \{C\} \cup \{res_B(C, B)\}$ этот процесс идёт до тех пор, пока либо Г не будет содержать 0, либо не найдётся дизъюнктов В и С указанного вида.

В первом случае указанное множество противоречиво, а во втором непротиворечиво.

Билет 12. Проблемы аксиоматического исчисления высказываний.

Исчисление высказываний для своего обоснования требует решения 4-х проблем:

1) проблемы разрешимости,
2) проблемы непротиворечивости,
3) проблемы полноты,
4) проблемы независимости.

<u>1. Проблема разрешимости исчисления высказываний.</u>

Проблема разрешимости исчисления высказываний заключается в поиске алгоритма, который позволил бы для любой заданной формулы исчисления высказываний определить, является ли она доказуемой или не является. В исчислении высказываний такой алгоритм существует.

Действительно, любая формула исчисления высказываний может рассматриваться как формула алгебры высказываний, и, следовательно, можно рассматривать ее логические значения на различных наборах значений входящих в нее переменных.

Это пример, взят не из лекций, а с постороннего ресурса.

Пусть А – любая формула исчисления высказываний, а $x_1, x_2, \dots, x_n$ – перечень входящих в нее переменных. Вычислим $R_{a1,a2,\dots,an}(A)$ на всех наборах значений $a_1, a_2, \dots, a_n$ входящих в нее переменных. Если при этом $R_{a1,a2,\dots,an}(A)=1$, на всех наборах $a_1, a_2, \dots, a_n$, то формула А – тождественно истинна, и, значит, по теореме о доказуемости тождественно истинной формулы она доказуема.

Если же существует набор значений переменных $a_1^0, a_2^0, \dots, a_n^0$ такой, что $R_{a_1^0 \dots a_n^0}(A) = 0$, то формула А – не тождественно истинная, и, значит, по теореме 1 §8 она не доказуема.

<u>2. Проблема непротиворечивости исчисления высказываний.</u>

Логическое исчисление называется непротиворечивым, если в нем не доказуемы никакие две формулы, из которых одна является отрицанием другой.

Иначе говоря, аксиоматическое исчисление называется непротиворечивым, если в нем не существует такая формула А, что доказуема как формула А, так и формула $\overline{A}$.

Если в исчислении обнаруживаются доказуемые формулы вида А и $\overline{A}$, то такое исчисление называется противоречивым.

Теорема: ИВ непротиворечиво.

Док-во: По теореме об общезначимых ф-х всякая выводимая формула тождественно истинна. Отрицание формулы не является тавтологией, следовательно, ни для какой формулы А не возможно, чтобы одновременно выводились А и $\overline{A}$.

3. Проблема полноты исчисление высказываний.

Определение 1.

Аксиоматическое исчисление высказываний называется полным в узком смысле, если добавление к списку его аксиом любой недоказуемой в исчислении формулы в качестве новой аксиомы приводит к противоречивому исчислению.

Определение 2.

Исчисление высказываний называется полным в широком смысле, если любая тождественно истинная формула в нем доказуема.

Теорема: ИВ полно в узком смысле.

Док-во: Пусть А – произвольная не выводимая формула. (по теореме об общезначимых формулах в качестве А можно взять любую опровержимую формулу.) < $x_1,\ldots,x_n$ >-список её переменных. Так как А – не выводимая формула, то существует такая оценка списка её переменных$< \alpha_1,\ldots,\alpha_n >$, что $A(x_1,\ldots,x_n)|_{\alpha_1,\ldots,\alpha_n}$ =Л

Пусть $B_1,\ldots,B_n$- любые тождественно истинные ф-лы, зависящие от переменных $x_1,\ldots,x_n$ Рассмотрим следующий набор $B_1^{\alpha_1},\ldots,B_n^{\alpha_n}$ где $B_i^{\alpha_i}=\{B_i, \text{если } \alpha_i = И; \overline{B_i}, \text{если } \alpha_i = Л\}$

Осуществим их подстановку в формулу А.

Заметим, что для любого набора $<б_1,\ldots,б_n> B_i(x_1,\ldots,x_n)|_{б_1,\ldots,б_n} \equiv И, т.к. B_i - \text{тавтология}$. Тогда $B_i^{\alpha_i}(x_1,\ldots,x_n)|_{б_1,\ldots,б_n}=\alpha_i$ и $A(B_1^{\alpha_1},\ldots,B_n^{\alpha_n})=A(\alpha_1,\ldots,\alpha_n)$=Л

Таким образом $A(B_1^{\alpha_1},\ldots,B_n^{\alpha_n}) \equiv Л$

Тогда отрицание $\overline{A(B_1^{\alpha_1},\ldots,B_n^{\alpha_n})} \equiv И$, получаем тавтологию, которая по теореме об общезначимых фор-х выводима в ИВ. С другой стороны если формулу $A(x_1,\ldots,x_n)$ добавить к списку аксиом исчисления, то она станет выводимой в новом исчислении как аксиома.

В новом исчислении $A(B_1^{\alpha_1},\ldots,B_n^{\alpha_n})$ получается в результате одновременной подстановки, то есть так же будет выводимой . Следовательно новое ИВ будет противоречивым, т.к. в нем одновременно выводятся А и $\overline{A}$.

4. Проблема независимости аксиом исчисления высказываний.

Заключается в невыводимости любой из аксиом из остальных аксиом по правилам вывода данной теории.

Теорема: Система аксиом ИВ независима.

Док-во: основано на некоторых интерпретациях переменных и логических операций исчислений. В простейшем случае допускается, что переменные в аксиомах могут принимать только 2 значения: 1 и 0.

Все логические операции кроме одной определяются так же, как в Алгебре высказываний $\overline{\ },\vee,\&,\rightarrow$.

Одну из логический операций определяют специально так, чтобы та аксиома, в которую эта операция входит и независимость которой доказывается, не являлась тождественно равная 1.

При такой интерпретации все аксиомы, кроме исследуемой принимают значение 1. Ясно, что если такая интерпретация возможна, то исследуемая аксиома не зависит от остальных, так как если бы она была выводима из остальных, то она, как и все формулы выводима из совокупности аксиом, кроме исследуемой, приняла бы единственное значение 1.

Билет 13 Определение предиката. Область определения, множество истинности предиката. Операции над предикатами, кванторы существования и всеобщности.

Определение. Предикат – это функция $P(x_1,x_2,\ldots,x_n)$, которая может принимать значения «Истина» (И) или «Ложь» (Л) , а x_i могут принимать значение из некоторой области М.

Пр. “х – четное число” –унарный предикат (одноместный), “х/у – четное число” – бинарный (двухместный)

Область определения. Множество M, на котором определен предикат - называется областью определения предиката.

Множество истинности – множество I_p на котором $P(x_1,x_2,\dots,x_n) \equiv И$

Если I_p=M, то предикат предикат $P(x_1,x_2,\dots,x_n)$ называется тождественно истинным,

Если I_p=ø, то предикат тождественно ложный.

Операции над предикатами

$P(x)\&Q(x)$	$P(x)VQ(x)$	$P(x)\sim Q(x)$	$P(x) \to Q(x)$	$P(x)$
$I_P \cap I_Q$	$I_P \cup I_Q$	$I_P = I_Q$	$I_P \subset I_Q$	$I_P = M \backslash I_P$

Кроме операций логики высказываний к предикатам применяются кванторы.

∃-квантор существования, ∀-квантор всеобщности.

$\exists x P(x)$ будем понимать высказывание истинным, когда существует элемент $x \epsilon M$, для которого $P(x)$ – истинно, и ложным в противоположном случае.

$\forall x P(x)$ будем понимать высказывание истинным, когда для каждого элемента $x \epsilon M$ - $P(x)$ – истинно, и ложным в противоположном случае.

В предикате $P(x)$ переменная x называется свободной, в высказываниях $\exists x P(x), \forall x P(x)$ x называется связанной.

Билет 14 Формулы логики предикатов, свободные и связанные переменные.

Алфавит логики предикатов содержит следующие символы:

1)Символы переменных высказываний P_1, Q_2, r_3

2)Символы предметных переменных и предметных констант $x_1, x_2, \dots, x_n$

3)Символы предикатов $A_1^{(t)} \dots A_n^{(t)}$, где t=0,1,2… кол-во предметных переменных от которых зависят предикаты

4)Функциональные символы $f_j^{m_j}$, где $m_j \in N$, - m_j местный функциональный символ

5)Логические символы $\&, V, \to, \sim \dots$

6)Символы кванторы $\exists, \forall$

7)Символы (,)

ОПР: Множество пердикатных букв вместе с множеством функциональных букв и констант называется сигнатурой языка данной теории.

ОПР: Термами языка ЛП являются 1)предикатные переменные и предметные константы 2) если f^n – n местный функциональный символ и $t_1, \dots, t_n$-термы, то $f^n(t_1, \dots t_n)$–тоже терм.

ОПР: Атомарной формулой сигнатуры называют выражение $P^n(t_1, \dots t_n)$, где P^n –n местный предикатный символ $t_1, \dots, t_n$-термы. все предметные переменные атомарной формулы свободные, связных нет.

Формулы логики предикатов. Слово в алфавите ЛП называется формулой если оно удовлетворяет след. индуктивному определению:

1)Атомарная формула – это формула

2)Если A и B – формулы, то $A \sim B, A\&B, A \to B, AVB$ тоже формулы, при условии что одна и та же переменная не является в A – свободной, а в B – связанной или наоборот.

3)Если A – формула, то (не A) тоже формула, свободные и связанные переменные совпадают

4)Если A(x) – формула, то $\exists A(x), \forall A(x)$ тоже формулы, причем если в A x-была совбодной, то в $\exists x A(x), \forall x A(x)$ x-связанная

*из этого определения видно что всякая формула алгебры высказываний является формулой ЛП.

ОПР: Подслово формулы A, которой само является формулой называется подформулой формулы A.

Пр. $\exists x P(x,y) \to \forall x P(x,y)$ y-свободная переменная, x-связанная

Билет 15 Равносильность формул в логике предикатов и в различных интерпретациях. Основные равносильности: перестановка кванторов и переименование связанных переменных.

Равносильность формул в логике предикатов и в различных интерпретациях.

Пусть формулы F и G имеют одно и то же множество свободных переменных (может пустое).

Опр: Формулы F и G равносильны в данной интерпретации, если на любом наборе значений свободных переменных они принимают одинаковые значения. (если в одной и той же интерпретации эти формулы выражают один и тот же предикат)

Опр: Формулы F и G равносильны на множестве M, если они равносильны во всех интерпретациях, заданных на M.

Опр: Формулы F и G равносильны (в Логике Предикатов) если они равносильны на всех множествах, F≡G.

Для формулы Логики Предикатов (ЛП) сохраняются все равносильности и правила равносильных преобразований логики высказываний, но имеются равносильности самой ЛП, связанные с кванторами.

Основные равносильности:

1. Перестановка кванторов (одноименных)

$\forall$ – квантор всеобщности

$\exists$ – квантор существования

$\forall x \forall y P(x,y) \equiv \forall y \forall x P(x,y)$

$\exists x \exists y P(x,y) \equiv \exists y \exists x P(x,y)$

Однако разноименные кванторы переставлять нельзя, пример:

1) $\forall y \exists x P(x,y)$

2) $\exists x \forall y P(x,y)$

Пусть $P(x,y) = (x > y)$ на мн-ве M=N*N.

Тогда 1е высказывание означает, что какое бы натуральное число мы не взяли, всегда найдется натуральное число еще больше этого. Это высказывание истинно.

2е высказывание означает, что существует самое большое натуральное число. Оно ложно на M.

2. Переименование связанных переменных

Заменяя связанную переменную формулы A другой переменной, входящей в эту формулу в кванторе и всюду в области действия квантора, получим формулу равнозначную A.

Билет 16. Правила переноса квантора через отрицание в формулах логики предикатов.

1) $\overline{\forall x \mathrm{A}(x)} \equiv \exists x \overline{\mathrm{A}}(x)$

2) $\overline{\exists x A(x)} \equiv \forall x \overline{A}(x)$

Докажем первый закон: пусть $x_1, x_2, \ldots, x_n$ – множество (может пустое) всех свободных переменных формулы A отличных от x. Пусть пара { M , f }-произвольная интерпретация.

Рассмотрим произвольный набор значений свободных переменных: $<a_1, a_2, \ldots, a_n>$

$a_i \in M$.

Исследуем какие логические значения на этом наборе примут формулы: $\overline{\forall x \mathrm{A}(x)}$ и $\overline{\exists x A(x)}$. Возможны два случая:

1) Для $\forall a \in M$: $A(x)|_{<a_1, a_2, \ldots, a_n>} = \text{И}$

2) Для $\forall a_0 \in M$: $A(x)|_{<a_0, a_1, a_2, \ldots, a_n>} = \text{Л}$

1. Для $\forall a \in M$: $\overline{A}(x)|_{<a_0, a_1, a_2, \ldots, a_n>} = \text{Л}$, отсюда по определению:

   $\exists x \overline{A}(x)|_{<a_0, a_1, a_2, \ldots, a_n>} = \text{Л}$, с другой стороны

   $\forall x A(x)|_{<a_1, a_2, \ldots, a_n>} = \text{И}$, т.к. в этом случае $\forall a \in M : A(x)|_{<a, a_1, \ldots, a_n>} = \text{И}$, отсюда

   $\overline{\forall x A(x)}|_{<a_1, a_2, \ldots, a_n>}$=Л, значит $\overline{\forall x A(x)} \equiv \exists x \overline{A}(x)$ выполняется в случае 1.

2. Для $\forall a_0 \in M$ $\overline{A}(x)|_{<a_0, a_1, a_2, \ldots, a_n>} = \text{И}$, отсюда $\exists x \overline{A}(x)|_{<a_1, a_2, \ldots, a_n>} = \text{И}$,

   с другой стороны в этом случае $\forall x A(x)|_{<a_1, a_2, \ldots, a_n>}$ =Л, следовательно

   $\overline{\forall x A(x)}|_{<a_1, a_2, \ldots, a_n>}$ =И, следовательно и во втором случае искомая равносильность доказана,

   значит она полностью доказана.

Билет 17. Правила выноса квантора за скобки в формулах логики предикатов.

Вынос квантора за скобки:

Постоянный предикат можно вносить под знак квантора и выносить из под него конъюнкции и дизъюнкции.
C& ∃ x B(x) ≡ ∃ x (C&B(x))
$C \vee \exists x\, B(x) \equiv \exists x(C \vee B(x))$
$C\&\forall x\, B(x) \equiv \forall x(C\&B(x))$
$C \vee \forall x\, B(x) \equiv \forall x(C \vee B(x))$
Док-во: *(1ой равносильности)*
Пусть x1…xn – все свободные переменные формулы $C\&\exists x\, B(x)$. Рассмотрим произвольную интерпретацию с множеством M. Пусть <a1,...,an>, ai∈M – произвольный набор свободных переменных x1,…,xn. Так как формула C не содержит переменные x, то можно определить ее значение.
Если $C|_{<a_1,\dots,a_n>} = Л$, то $(C\&\exists B(x))|_{<a_1,\dots,a_n>} = Л$ и для $\forall a \in M$ на наборе значений <a, a1,…,an> свободных переменных <x, x1,…,xn> формула C&B(x) принимает значение Л. Отсюда $\exists x(C\&B(x))|_{<a_1,\dots,a_n>} = Л$. Если же $C|_{<a_1,\dots,a_n>} = И$, то для $\forall a \in M$ на наборе <a, a1,…,an> формулы C&B(x) и B(x) принимают одинаковые истинностные значения.
Следовательно, $(C\&\exists x B(x))|_{<a_1,\dots,a_n>} = \exists x B(x)|_{<a_1,\dots,a_n>} = \exists x(C\&B(x))|_{<a_1,\dots,a_n>}$
Таким образом, исходная равносильность полностью доказана.
Замечание: если оба предиката содержат переменную x, то выполняется лишь 2 равносильности из четырех: $\forall x(A(x)\&B(x)) \equiv \forall x A(x)\&\forall x B(x)$ и $\exists x(A(x) \vee B(x)) \equiv \exists x A(x) \vee \exists x B(x)$.

Билет 18 Нормальные формы логики предикатов. Теорема о ПНФ.

Формулы(ф-лы), в которых из логических символов имеются только символы &,∨, ¬, причем ¬ встречается только над символами предикатов, называются нормальными формами (НФ).
Используя равносильности алгебры высказываний и логики предикатов (ЛП) каждую ф-лу ЛП можно привести к НФ.
Для ∀ ф-лы ЛП ∃ равносильная ей НФ, причем множество свободных и связных переменных этих ф-л совпадают (без док-ва).
Среди НФ особое значение имеют предваренные нормальные формы (ПНФ).
ПНФ формулы ЛП называется такая НФ, в которой либо полностью отсутствуют кванторные операции, либо в последовательности слов, образующих ф-лу, кванторы предшествуют всем остальным символам $(\partial x_1)(\partial x_2)\dots(\partial x_n)A(x_1,x_2,\dots,x_m)$,
где $\partial x_i \equiv \forall x_i$ или $\exists x_i$. В ф-ле A кванторов нет. Смысл записи ф-лы в ПНФ состоит в том, чтобы разделить ф-лу на 2 части: кванторную и бескванторную. В таком виде ее проще анализировать.
1)$\forall x \exists y(\bar{A}(x) \vee B(x,y))$ – ПНФ
2)$\forall x \bar{A}(x) \& \exists y B(y)$ - НФ, не являющаяся предваренной
Если все ∂x_i - кванторы всеобщности(∀), то ф-ла называется универсальной - $\partial x_i \equiv \forall x_i$
Если же все $\partial x_i \equiv \exists x_i$, то существующая ф-ла.
Если ∃ i (0≤i≤n) такое, что $\partial_1, \partial_2, \dots, \partial_i$ - кванторы существования(∃), а $\partial_{i+1}, \partial_{i+2}, \dots, \partial_n \equiv \forall$, то ф-ла называется существующей универсальной ф-лой или скулемовской.
Теорема о ПНФ
Всякая форма ЛП может быть приведена к равносильной ей ПНФ.
Идея док-ва:
Если ф-ла содержит отрицания над кванторами, то с помощью 1ого закона ЛП отрицание выносится под знак квантора. Если ф-ла имеет вид $A_1 \vee A_2$ или $A_1 \& A_2$ тогда возникает необходимость сначала переименовать в A_2 связные переменные так, чтобы все они в A_1 и A_2 были различны, а затем воспользоваться 2ым законом ЛП.
Пример. Привести к ПНФ
$\overline{P \to \exists x R(x)} \equiv \overline{\bar{P} \vee \exists(x) R(x)} \equiv P\&\overline{\exists x R(x)} \equiv P\&\forall x \bar{R}(x) \equiv \forall x(P\&\bar{R}(x))$

Билет 19. Выполнимость и общезначимость для предикатов. Основные общезначимые формулы в логике предикатов.

А выполнима в данной интерпритации если существует набор $< a_1 \dots a_n >$ где $a_i \in M$ значений свободных переменных $x_1 \dots x_n$ что $A|_{<a_1 \dots a_n>} = U$.
Формула А истинна в заданной интерпритации если она принимает значение И на ∀ оценке списка$< a_1 \dots a_n >$ где $a_i \in M$ знач. своб перем-х $x_1 \dots x_n$.
Формула А общезначима (тождественно истинна в лп) если она истинна в каждой интерпритации.
Говорят что А выполнима в лп, если∃ интерп в которой она выполнима.
Утв1(об общезн-ти) общезн-й явл ф-ла $\forall x A(x) \to A(y)$ (1) где у не входит в ф-лу A(x)

Док-во: пусть $x_1 \dots x_n$ все своб перем-е в-лы A(x) => $y, x_1 \dots x_n$ все переменные формулы (1)
1) рассмотрим произвольную с множеством M. пусть$b, a_1 \dots a_n$ где $b \in M, a_i \in M$ произв набор значений всех свободных переменных формулы(1)
Покажем: $\forall x A(x) \to A(y)|_{<b,a_1 \dots a_n>} = И$ Для $A(x)$ либо $\exists a_0 \in M$: $A(x)|_{<a_0,a_1 \dots a_n>} = И$ либо $\forall a \in M$(в тч для a=b): $A(x)|_{<a_0,a_1 \dots a_n>}$=И если $A(x)|_{<a_0,a_1 \dots a_n>}$=Л то $\forall x A(x)|_{<a_1 \dots a_n>} = Л$ тогда $\forall x A(x) \to A(y)|_{<b,a_1 \dots a_n>} = И$ 2) $\forall x A(x)|_{<a_1 \dots a_n>} = И$ $A(y)|_{<b,a_1 \dots a_n>} = И$ тогда $\forall x A(x) \to A(y) \equiv И$ тк$И \to И \equiv И$=> (1) общезначимая
Утв2(об общезначимости) $A(y) \to \exists x A(x)$ (2), где у не входит в A(x) - общезначимая
Док-во: в силу утв1 (1)-общезн=> заменим A на
$\bar{A}$: $\forall x \bar{A}(x) \to \bar{A}(y) \equiv \overline{\forall x \bar{A}(x)} v \bar{A}(y) \equiv \exists x A(x) \to \bar{A}(y) \equiv A(y) \to \exists x A(x)$=>(2) общезначима.
Замечание:
Тк одноим-е кванторы можно переставлять то эквивалентны и $\exists x \exists y A(xy) \sim \exists y \exists x A(xy)$
$\forall x \forall y A(xy) \sim \forall y \forall x A(xy)$
Док: Равносильности ф-л ЛП требует либо детального опред значений ф-л либо использования известных равносильностей.
Для определения общезн-ти ф-лы лп иногда не достаточно использ-я равнос-й. отличить решения этой задачи помогут т. Об общезначимости

Билет 20. Теоремы об общезначимости и выполнимости в логике предикатов. Проблема разрешимости в общем случае (теорема Черча) и для формул, содержащих только одноместные предикатные символы.

Теорема 1.
Формула А общезначима согда (¬А) невыполнимо (всюду тождественно ложно.)
Доказательство

- ⇨ Пусть А –общезначима, тогда (¬А) тождественно ложная на всякой области М, следовательно она не является выполнимой ни к в какой области М.

<= Тоже самое, в другую сторону.

Теорема 2
Формула А выполнима согда формула (¬А) не общезначима.
Доказательство вытекает из того, что если формула А выполнима, то для нее существуют 2 области М1 и М2 в одной из которых она истина, а в другой ложна.

- ⇨ Если взять ту область, где А-истинна, то (¬А) там будет ложна.

<= Тоже самое, в другую сторону.

Теорема 3
Если функции А и В равносильны в ЛП, то формула А~В – общезначима.
Доказательство: следует из определения операции равносильности и ~.
Проблема разрешимости: указать эффективный алгоритм распознания общезначимости формул.
Метод перебора всех вариантов для такой задачи в общем случае не применим, т.к. вариантов может быть бесконечно много.
Теорема Черча.
Не существует алгоритма, который для произвольной формулы ЛП устанавливал бы одназначна она или нет.
Эта проблема решается лишь в некоторых частных случая, например соответствующий алгоритм существует для формул ЛП, содержащих только одноместные предикатные символы. Логика, в которой используются только одноместные предикаты практически соотвутствует логике Аристотеля. Здесь кванторные операции можно заменить операциями V,&… и тем самым свести формулу ЛП к формулам ЛВ, для которой проблема разрешимости имеет алгоритмическое решение.
Алгоритм проверки таких формул осуществляется с помощью теоремы:
Критерии выполнимости ФЛП.
Пусть F формула, содержащая ровно n предикатных символов. F выполнима согда она выполнима во всех интерпритациях с множества М, содержащем не более 2^n символов.
Кроме того, можно проверить общезначимость ФЛП у которой в ПНФ содержаться кванторы одного типа.

В любом случае, пытаться проверить общезначимость формулы целесообразно после приведения этой формулы к ПНФ.

Билет 21. Язык, система аксиом и основные правила вывода исчисления предикатов.

Исчисление предикатов определяется следующим образом:

1. Алфавит ИП состоит из символов логики предикатов и знака следования «|-»
2. Формула ИП определяется так же как и в логике предикатов с поправкой на знак «|-». Внимание надо обратить на:

1) В формуле свободные и связанные переменные обозначаются разными буквами.
2) Если квантор находится в области действия другого квантора, то переменные связанные этими кванторами обозначаются разными буквами.

3. Аксиомы ИП делятся на 2 группы:

1) Одна из систем аксиом исчисления высказываний (т.к. ИП это расширение ИВ)
2) Касается предикатов и состоит из 2ух аксиом:

(P1) $\forall$ x F(x)→F(y) - аксиома общности

(P2) F(y)→ $\exists$x F(x) – аксиома введения квантора существования.

4. Правила вывода

1) «не разобрал слово» как в ИВ
2) Правило обобщения: то $\rightarrow \frac{F \rightarrow C(x)}{F \rightarrow \forall x C(x)}$
3) Правило введения квантора существования: $\left.\frac{C(x) \rightarrow F}{\exists x C(x) \rightarrow F}\right\}$ C(x) – содержит свободные вхождения x, F- не содержит.

Билет 22. Производные правила вывода в исчисление предикатов: правила переименования связанных переменных, правило связывания квантором.

Правило переименования связанных переменных

1) $\forall x F(x) | - \forall y F(y)$
2) $\exists x F(x) | - \exists y F(y)$

F(x) не содержит свободных вхождений y,но содержит свободные вхождения x, ни одно из которых не входит в область действия квантора по y.

Доказательство:

1) $\forall x F(x)$ – гипотеза
2) По аксиоме (P1) справедлива формула $\forall x F(x) \rightarrow F(y)$
3) К шагу 2 применим правило обобщения $| - \forall x F(x) \rightarrow \forall y F(y)$
4) MP для (1) и (3): $\forall x F(x) | - \forall y F(y)$

Для квантора существования доказать самостоятельно!

Правило связывания кванторов.

$F(x) | -_K \forall x F(x)$

1) F(x) – гипотеза по условию
2) По т.2 ИВ $A | -_L B \rightarrow A$, то есть формула $B \rightarrow A$ доказуема для любого B, в том числе и для доказуемов формуле B. Значит $F(x) | -_K B \rightarrow F(x)$
3) По правилу обощения $\frac{B \rightarrow F(x)}{B \rightarrow \forall x F(x)}$, B не содержит свободных вхождений x.
4) Из (3) по правилу заключения MP (для B и $B \rightarrow \forall x F(x)$) следует доказуемость $F(x) | -_K \forall x F(x)$.

Билет 23. Теоремы об общезначимых формулах и о замене эквивалентных подформул в исчислении предикатов.

Теорема об общезначимых формулах

$\vdash_k$ F $\Leftrightarrow$ F – теоремами ИП являются общезначимые формулы и только они.

Док-во:

1) → Аксиомы ∀ системы – по сути тавтологии. Все правила вывода сохраняют общезначимость, т.е. их применение к общезначимым формулам снова даёт общезначимые формулы, значит теоремы ИП – общезначимые формулы.
2) ← Доказал впервые Гёдель. Оно сложнее и не приводится в этом курсе.
Опр. Две формулы F и G эквивалентны, если $\vdash F \sim G$
Из теоремы об общезначимых формулах ИП следует, что между соотношениями ЛП и формальными эквивалентностями в ИП имеются аналогичные соответствия.
$F \sim G \equiv F \to G$.
Теорема о замене эквивалентных подформул
Пусть F(A) – формула с вхождением A. F(B) – формула, полученная из F(A) заменой этого вхождения A формулой B. Тогда, если $\vdash A \sim B$, то $\vdash F(A) \sim F(B)$. Благодаря этому правилу можно получить доказуемые эквивалентности не строя их непосредственного вывода.

Билет 24. Наиболее важные эквивалентности исчисления предикатов и их применение для построения предваренной нормальной формы.

A и B формулы, не содержащие свободных вхождений x.

1) $\forall x(F(x)\&G(x)) \sim (\forall x(F(x)\& \forall xG(x))$
2) $\exists x(F(x) \vee G(x)) \sim (\exists x(F(x) \vee \exists xG(x))$
3) $A\& \forall xF(x) \sim \forall x(A\&F(x))$
4) $A\& \exists xF(x) \sim \exists x(A\&F(x))$
5) $A \vee \forall xF(x) \sim \forall x(A \vee F(x))$
6) $A \vee \exists xF(x) \sim \exists x(A \vee F(x))$
7) $A \to \forall xF(x) \sim \forall x(A \to F(x))$
8) $A \to \exists xF(x) \sim \exists x(A \to F(x))$
9) $\forall xF(x) \to B \sim \exists x(F(x) \to B)$
10) $\exists xF(x) \to B \sim \forall x(F(x) \to B)$

Из 10 данных эквивалентностей первые 6 аналогичны закону (2) ЛП, остальные 4 доказываются с помощью закона (1) ЛП и равносильности ЛВ. Используя данные 10 эквивалентностей в формулах ИП можно выносить кванторы за скобки. С помощью правил переноса квантора через отрицание, а также правила переименования переменных кванторы можно выносить за скобки в $\forall$ формуле. При этом получится предварённая нормальная форма.

Билет 25 Проблемы аксиоматического исчисления предикатов.

Исчисление предикатов для своего обоснования требуют решения 4ёх проблем: разрешимости, непротиворечивости, независимости и полноты.

1ая проблема: (разрешимости) заключается в поиске алгоритма, который бы для любой заданной формы исчисления определял был – является ли она доказуемой в этом исчислении или нет. В исчислении предикатов разрешающийся алгоритм построить невозможно из-за бесконечности предельной области, который приводит в общем случае к бесконечным таблицам истинности. Доказательство этого факта стало возможным лишь после появления точного определения алгоритма.

2ая проблема: (не противоречивость исчисления), то есть не существует в этом исчислении формулы A такой, которая была бы доказуема с не A.

Теорема: Исчисление предикатов не противоречиво.

Док-во: Допустим И.П. противоречиво, тогда в нём выводима всякая формула, в частности формула A, состоящая из 1ой буквы, но тогда A была бы выводима и в И.В. , что неверно по теореме об общезначимых формулах И.В.

3я проблема: (независимости системы аксиом исчисления предикатов). Система аксиом И.П. – независимая система ,то есть в этой системе нет лишних аксиом. Эту независимость можно установить, например сведением к вопросу о непротиворечивости данной системы.

4я проблема: (полноты). Аксиоматическое исчисление называется полном в узком смысле, если добавление к списку его аксиом любой, не доказуемой в этом исчислении формулы, в качестве новой аксиомы, приводит к противоречивому исчислению. Исчисление предикат оказывается не полным в узком смысле. К его аксиомам можно присоединить без противоречия недоказуемую в нём формулу. Аксиоматическое исчисление называется полном в широком смысле слова, если любая тождественно истинная формула в нём доказуема . Исчисление предикат полно в широком смысле по теореме Гёделя. В

И.П. нельзя вывести сколь ни будь содержательное по существу высказывание, в частности математическое.

Билет 26 Формализация понятия алгоритма.

Алгоритм – общий, единообразный, точно установленный способ решения любой задачи из данной массовой проблемы (бесконечного множества однотипных задач). Задачи одной и той же массовой проблемы отличаются друг от друга лишь значением входящих в них параметров. Тем не менее даже при таком интуитивном понятии можно выделить некоторые характерные черты алгоритма: 1) Дискретность (состоит из действий, выполняемых по шагам).2)Детерменированность (между всеми величинами, получаемыми алгоритмом жёсткая причинная связь.3)Массовость (начальная система величин выбирается из некоторого множества).4)Результативность (остановка алгоритмического процесса, после конечного числа шагов, с указанием достигнутого конструктивного объекта в виде результата). Интуитивное понятие алгоритма работает, когда речь идёт о найденном алгоритме решения конкретной проблемы. Однако строго математически доказать не существование алгоритма, пользуясь лишь его интуитивным определением невозможно. Для этого нужно формальное определение алгоритма. Тьюринг А.М. предложил определение в виде некоторой машины, теперь называемой машиной Тьюринга. Э.Л. Пост предложил определение алгоритма в виде некоторого набора правил, называемых системой продукций. А.Чёрч, С.К. Клиник, опираясь на более ранние работы Ж.Эрбрана. К.Гёдель предложил определение алгоритма рекурсивными функциями. Наиболее широкое определение А.Н. Камогова.

Типы алгоритмических моделей:

1) Рекурсивные функции. Этот тип связывает понятие алгоритма с числовыми функциями, заданными на множестве натуральных чисел и принимающие значения на том же множестве.
2) Машины Тьюринга. Связывает понятия алгоритма с механическим устройством. Способна выполнять дискретно элементарные действия над элементарными объектами.
3) Нормальный алгоритм Маркова. Связывает понятия алгоритма с классом словарных преобразований, в результате замены части слов или всего слова другим словом.

Билет 27. Понятие рекурсивных функций. Примитивно рекурсивные функции: базовые функции и элементарные операции.

Рекурсивным заданием функции называется способ определения значения в некоторой точке, через известные значения этой функции в предшествующих точках.

Заданные рекурсивно числовые функции $f: M \to N_0$, где $M \subseteq N_0^n$ будем называть рекурсивными.

Примитивно рекурсивные функции.

Базовые фукции.

Простые одношаговые рекурсивные функции называются базовыми.

1)Нуль функция 0(x)=0

2)Функция тождества(проектирующая функция, функция введения фиктивных переменных)

$I_m^n(x_1, \dots, x_n) = x_m$, где $1 \le m \le n$, если $n = 1$, то $I(x) = x$

3)Функция следования(прибавление единицы) $\lambda(x) = x + 1,\ x' = x + 1$

*эта функция позволяет по числу построить сколь угодно большой начальный отрезок множества натуральных чисел

Элементарные операции.

Операции, с помощью которых можно получить из базовых функций рекурсивные называются элементарными.

1) Операция суперпозиции

Говорят, что n-местная функция $\varphi(x_1 \dots x_n)$ получена с помощью операторов суперпозиции из m-местной функции $\varphi(x_1 \dots x_m)$ и n-местных функций $f_1(x_1 \dots x_n)$, …,$f_m(x_1 \dots x_n)$, если

$\varphi(x_1 \dots x_n) = \varphi(f_1(x_1 \dots x_n), \dots, f_m(x_1 \dots x_n))$

2) Операция примитивной рекурсии

Говорят, что n+1-местная функция $f(x_1,\dots,x_n,y)$ получена из n-местной $\varphi(x_1,\dots,x_n)$ и n+2- местной $\psi(x_1,\dots,x_n,y,z)$ с помощью операции примитивной рекурсии если ее значения можно вычислить по формуле:

$$\begin{cases} f(x_1 \dots x_n, 0) = \varphi(x_1,\dots,x_n) \\ f(x_1,\dots,x_n,y+1) = \psi(x_1,\dots,x_n,y,f(x_1,\dots,x_n,y)) \end{cases}$$

При n=0 схема примитивной рекурсии имеет вид

$$\begin{cases} f(0) = a \quad a = const \\ f(y+1) = \psi(y,f(y)) \end{cases}$$

Будем говорить что функция получена из функции с помощью итераций $f(x) = i(g(x))$ если:

$$\begin{cases} f(0) = 0 \\ f(x+1) = g(f(x)) \end{cases}$$

Билет 28. Примеры простейших примитивно рекурсивных функций.

1) Базовые функции
2) $0(x_1,x_2,\dots,x_n) = 0$ – ПРФ, так как $0(x_1,x_2,\dots,x_n)$=$0(I_n^1(x_1),\dots,I_n^n(x_n))$, т.е. получаем из базовых функций 0(x) и I_n
3) f(x)=x+n ПРФ т.к. f(x)=x (прибавили единицу n раз)
4) $f_+(x,y) = x+y$ – ПРФ т.к. ее можно получить из функций тождества и следования с помощью оператора примитивной рекурсии.

   $f_+(x,0) = x+0 = I_2^1(x,y)$

   $f_+(x,y+1) = x+(y+1) = (x+y)+1 = f_+(x,y)+1 = \lambda(x+y)$

   $$\begin{cases} \varphi(x) = x \\ \psi(x,y,z) = z+1 \end{cases}$$
5) $f_*(x,y) = x*y$ – ПРФ т.к. ее можно получить из 0 и сложения с помощью оператора примитивной рекурсии.

   $f_*(x,0) = 0 = 0(x)$

   $f_*(x,y+1) = x*y+x = f_*(x,y)+x = f_+(f_*(x,y),x)$

   $$\begin{cases} \varphi(x) = 0 \\ \psi(x,y,z) = z+x \end{cases}$$
6) Поскольку ПРФ должны быть определены на всем N, то вместо обычной разности в теории рекурсивной функции вводят арифмитическую или усеченную разность:

   x-y (всюду ниже в этом билете над минусом ставится точка)

   x-y=$\begin{cases} x-y, & x \geq y \\ 0, & x < y \end{cases}$

   функция x-1

   $$\begin{cases} 0-1 = 0(x) \\ (y+1)-1 = y = I_2^2(x,y) \end{cases}$$

   $f_-(x,y) = x-y$

   $f_-(x,0) = x-0 = I_2^1(x,y)$

   $f_-(x,y+1) = x-(y+1) = (x-y)-1 = f_-(x,y)-1$

   $$\begin{cases} \varphi(x) = x \\ \psi(x,y,z) = z-1 \end{cases}$$
7) Сигнум

   sgn(x)=$\begin{cases} 0, & x = 0 \\ 1, & x > 0 \end{cases} \leftrightarrow \begin{cases} sgn(0) = 0 \\ sgn(y+1) = 1 \end{cases}$

   sgn(x,y)=$\begin{cases} 0, & x \leq y \\ 1, & x > y \end{cases}$ => sgn(x,y)=sgn(x-y)

   К этим двум функциям можно ввести дополнительные:

$\overline{\text{sgn}(x)} = 1 - \text{sgn}(x) = \begin{cases} 1, x = 0 \\ 0, x > 0 \end{cases}$

$\overline{\text{sgn}(x,y)} = 1 - \text{sgn}(x,y) = \begin{cases} 1, x \le y \\ 0, x > y \end{cases}$

Билет 29. Теорема а примитивной рекурсивности суммы и произведения примитивно рекурсивной функции. (без доказательства). Примитивная рекурсивность функции "частное от деления *x* на *y*", "остаток от деления *x* на *y*", "признак деления *x* на *y*".

Теорема. Пусть f – n – местная ПРФ, тогда также являются примитивно рекурсивными. :

$S(x_1,x_2 \dots x_n) = \sum_{i=0}^{x_n} f(x_1, \dots, x_{n-1}, i)$

$P(x_1,x_2 \dots x_n) = \prod_{i=0}^{x_n} f(x_1, \dots, x_{n-1}, i)$

(Без доказательства).

Примитивная рекурсивность функции "частное от деления *x* на *y*", "остаток от деления *x* на *y*", "признак деления *x* на *y*".

1) Частное от деления $\left[\frac{x}{y}\right] = \sum_{i=1}^{x} \overline{sgn}(iy - x)$

   x=7 y=3 $\Rightarrow \left[\frac{x}{y}\right] = 2$ и $\sum_{i=1}^{x} \overline{sgn}(3;-7)$

   По теореме о примитивной рекурсивности суммы произведения примитивно рекурсивной функции – ПРФ.

2) Остаток от деления $\frac{x}{y}$. $rest(x,y) = x - \left(y\left[\frac{x}{y}\right]\right) - \text{примитивно} - \text{рекурсивная ф} - \text{я}.$

3) Говорят,что

   $rest(x,y) = 0 => x \vdots y, \text{введем отдноместную функцию } div(x,y) =$

   $= \begin{cases} 1, \ rest(x,y) = 0 \\ 0, \ rest(x,y) > 0 \end{cases}$

div(x,y)=$\overline{sgn}$(rest(x,y)) => ПРФ

Билет 30. Ограниченный оператор минимизации и его применения. Теорема Робинсона об одноместных примитивно рекурсивных функциях (без доказательства).

Оператор минимизации.

В теории рекурсивных функций важную роль играет действие нахождения данной функции φ(z) значения z, которая является наименьшим корнем уравнения φ(z)=0. Выполняющий это действие оператор является оператором минимизации или μ - оператор. Существует несколько разновидностей.

Сначала рассмотрим ограниченный μ-оператор.

Ограниченный μ-оператор равен наименьшему значению y, удовлетворяющему условию $y \le z \& f(x_1 \dots x_{n-1}, y) = 0 (1)$ если такое y существует.

Замечание : Ограниченный μ-оператор по данной n-местной функции f строит новую функцию, значение которой равно наименьшему y, удовлетворяющему условию (1) . Следствия описания оператора необходимо дополнить еще 1 условием. Пример ф-я $[z] = \mu y_{y \le z}[\overline{sgn}(y+1)^z]$

Теорема Робинсона (1911-1995) об одноместных примитивно – рекурсивных функциях. Все одноместные ПРФ могут быть получены из функций $\lambda(x) = x + 1$ и $q(x) = x - \left[-\sqrt{x}\right]^2$- конечным числом операции суперпозиции итераций, сложения функций.

Билет 31. Неограниченный оператор минимизации. Частично рекурсивные функции(ЧР). Тезис Черча о вычислимых функциях.

Неограниченный оператор минимизации

Переход f к g называется неограниченным μ-оператором минимизации и обозначается :

$g(x_1 \ldots x_n)=\mu_y[f(x_1 \ldots x_n,y)=b]$

Частично рекурсивные функции. Определение

Функция называется частично рекурсивной если она базовая или может быть получена исходя из базовой функции, конечным числом применения операторов суперпозиции примитивной рекурсии и неограниченного оператора минимизации.

Замечание:

1) Из определений ПРФ и ЧРФ следует что множество ПРФ содержится в множестве ЧРФ. ПРФ⊂ЧРФ
2) Ограничения $y \le z$ в ограниченном μ- операторе даёт гарантию окончания вычислений, поскольку оно ограничивает сверху число вычислений корня уравнения :

$$f(x1, \ldots,xn-1, y) = 0$$

Такое ограничения является существенной особенностью примитивно рекурсивных функций, а не ограниченный рекурсивный μ оператор такими свойствами не обладает, а потому не является примитивно - рекурсивной

Тезис Черча

Всякая вычислимая функция является частично-рекурсивной.

В формулировке тезиса входит интуитивное понятие вычислимости, поэтому его нельзя доказать в обще принятом математическом смысле, опровергнуть его можно построив хотя бы один контр пример т.е. интуитивно вычислимую функцию не являющуюся ЧР. До сих пор примера построено не было и скорей всего их не существует!

Из этого тезиса следует, что всякая всюду определяемая вычисляемая функция является общерекурсивной. С помощью точного определения частичной рекурсивности можно доказывать или опровергать вычислимость. Существуют и другие уточнения понятия вычислимости функцию. Например понятие вычислимости по Тьюрингу.

Билет 32. Общерекурсивные функции. Функция Аккермана. Теорема Аккермана (без доказательства).

Всюду определённая ЧРФ называется общерекурсивной

ОРФ ⊂ ЧРФ

Из этих двух возникает гипотеза:

ПРФ ⊂ЧРФ
ОРФ ⊂ЧРФ } ⇨ ПРФ=ЧРФ-?

По аналогии с тем что sgn^{-1} (x) является ЧРФ, но не является ПРФ. Есть ОРФ-ии которые не являются ПРФ-ми.

Функция Аккермана

Первый пример такой функции (всюду определяемая и вычислимая) придумал немецкий математик Аккерман. Идея заключалась в построении последовательности функций B(x, y) каждая из которых растёт существенно быстрее предыдущей и создание с помощью этой последовательности функции A(x) которая растёт быстрее чем любой ПРФ-ций.

Построение функции Аккермана

$\varphi_0(a,y)=a+y$
$\varphi_1(a,y)=a*y$
$\varphi_2(a,y)=a^y$
, тогда

$\varphi_0(a,0)=a$ $\quad\quad$ $\varphi_0(a,1)=a+1$
$\varphi_1(a,0)=0$ $\quad\quad$ $\varphi_1(a,1)=a$ $\quad\quad$ (2)
$\varphi_2(a,0)=1$, $\quad\quad$ $\varphi_2(a,1)=a$

$\varphi_1(a,y+1)=a+ay=\varphi_0(a, \varphi_1(a,y))$
$\varphi_2(a,y+1)=a^{ay}=\varphi_1(a, \varphi_2(a,y))$

Продолжим эту последовательность, положим по определению

$$\varphi_{n+1}\begin{cases}(a,0)=1\\ \varphi_{n+1}(a,1)=a \quad\quad (3)\\ \varphi_{n+1}(a,y+1)=\varphi_n(a,\varphi_{n+1}(a,y))\end{cases}$$

Заметим что схема (3) согласуется с равенством (2) для $\forall n \in N$
(случай n=0 исключаем)

Схема (3) имеет вид примитивно рекурсивной следовательно все функции φ_nпримитивно рекурсивны растут они крайне быстро

Например:

φ_3(a,1)=a
φ_3(a,2)= φ_2(a,a)=a^a
φ_3(a,3)=a^{a^a} и т.д

Зафиксируем теперь значения a. Получим последовательность функций:

φ_0(2,y) , φ_1(2,y),….
$B(x.y) = \varphi_x$ (2,y)=2+y
$B(1,y) = \varphi_1$(2,y)=2y и т.д.

Кроме того определим диагональную функцию A(x)=B(x,x)

Из соотношения (3) следует:

B(0,y)=2+y

(4) B(x',0)=sgnx x'=x+1

B(x', y')=B(x,B(x', y)) y'=y+1

Соотношение(4) позволяет вычислить значение функции B(x,y), а следовательно и функции A(x). При чём при вычислении функции в данной точке. Нужно обратиться к значению функции в предыдущеё точке, как в примитивной рекурсии только здесь рекурсия ведётся по 2ум переменным (называется двойной или рекурсией второго порядка) , и это существенно усложняет характер упорядочивания точек, следовательно и понятие предшествования точек так же усложняется.

Теорема Аккермана

Функция Аккермана растёт быстрее чем любая ПРФ и следовательно не является примитивно рекурсивно функцией: более точно:

Для любой 1местной ПРФ f(x) $\forall$n:

$\forall x \geq n$ A(x)>f(x)

Поскольку функция Аккермана всюду определена, то она является частным случаем ЧРФ, а именно ОРФ

Билет 33. Словарные функции. Определение машины Тьюринга.

Основная идея: имитация алгоритмических процессов с помощью абстрактной математической машины(т.е. машины Тьюринга). Эта машина за конечное число шагов из исходных числовых данных в соответствии с заданными правилами может получить искомый числовой результат. Такая модель была предложена А.М. Тьюрингом в 1936 году.

Тезис Тьюринга:

Всякий алгоритм может быть задан некоторой функциональной схемой и реализован на соответствующей машине Тьюринга.

Словарные функции.

Пусть A -набор букв(алфавит),то A={a1,...an}.Такая конечная последовательность букв этого алфавита называется **словом**.

α =a1,…,an, α -слово, a1,…,an-буквы, ai$\in$A, $|\alpha|$-длина слова(число букв), λ -пустое слово, т.е. $|\lambda|$=0.

Свойства слов.

1) $\lambda\ \alpha = \alpha\ \lambda = \alpha$,

2) Закон ассоциативности: $(\alpha\ \beta)\gamma = \alpha(\beta\ \gamma)$,

3) Отсутствие коммутативности: $\alpha\ \beta \neq \beta\ \alpha$

A^n – множество всех слов длиной n, A* – множество всех слов, из букв алфавита A : $A^* = \bigcup_{n=0}^{\infty} A^n$,

a^n = a*a*…*a- n букв a, заметим, что $0^0 = 1^0 = \lambda$.

Слово, состоящее из бесконечного числа букв, называется **сверхсловом**.

A^ω – множество всех сверхслов, состоящие из букв алфавита A.

Отображение f: $A^* \to B^*$ – **словарная функция**.

Билет 34. Способы задания машин Тьюринга. Реализация на машине Тьюринга программы “перенос нуля”.

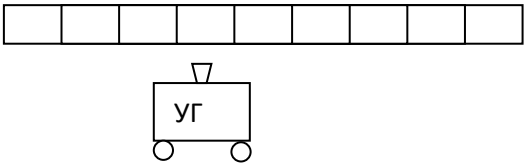
Машина Тьюринга включает в себя:

1. ***Внешний алфавит*** - конечное множество символов $A = \{a_0, a_1, a_2, \ldots, a_n\}$. В этом алфавите в виде слова кодируется та информация, которая подается в машину. Машина

перерабатывает информацию, поданную в виде слова, в новое слово. Обычно символ a_0 обозначает пробел.

Внешний алфавит - конечное множество символов

2. ***Внутренний алфавит*** - конечное множество символов $Q = \{q_0, q_1, q_2, \ldots, q_m\}$. Для любой машины число состояний фиксировано. Два состояния имеют особое назначение q_1 - начальное состояние машины, q_0 - заключительное состояние (стоп-состояние).
3. ***Бесконечная лента*** Бесконечная лента характеризует память машины. Она разбита на клеточки. В каждую клеточку может быть записан только один символ из внешнего алфавита.
4. ***Управляющая головка.*** Управляющая головка (УГ) передвигается вдоль ленты и может останавливаться напротив какой-либо клетки, т. е. считывать символ.



Работа машины складывается из следующих один за другим тактов, по ходу которых происходит преобразование начальной информации в промежуточные информации (к концу каждого такта совокупность знаков, хранящихся на ленте, образует соответствующую промежуточную информацию). В качестве начальной информации на ленту можно подать любую конечную систему знаков внешнего алфавита (любое слово в этом алфавите), расставленную произвольным образом по ячейкам.
В каждом такте работы машины она действует по функциональной схеме, которая имеет вид:

$$aiqj \Rightarrow avДqs$$

Здесь *ai,av* – буквы внешнего алфавита; *qiqs* – состояния машины; $Д \in \{R,L,C,P,E\}$.
Символ стирания Е подразумевает запись в данную ячейку пустого символа.
Перенос нуля.

$q_1 001^x 0 \to q_0 01^x 00$

При табличном представлении строки описывают текущее состояние МТ, столбцы, содержимое обозреваемой ячейки, а клетками этой таблицы являются правые части команд для соответствующей пары текущего состояния МТ.

Команда	Результат
$q_1 0 \to q_2 R$	$0\, q_2 01^x 0$
$q_2 0 \to q_3 1$	$0\, q_3 11^x 0$
$q_3 1 \to q_3 R$ x+1 раз	$011^x\, q_3 0$
$q_3 0 \to q_4 L$	$01^x\, q_4 10$
$q_4 1 \to q_5 0$	$01^x\, q_5 00$
$q_5 0 \to q_6 L$	$01^{x-1} q_э 100$
$q_6 1 \to q_6 L$ x раз	$q_6 01^x 00$
$q_6 0 \to q_0 0$	$q_0 01^x 00$

Билет 35. Неприменимость машины Тьюринга к исходной информации (привести пример). Тезис Тьюринга. Теорема о соответствии между частично рекурсивными функциями и функциями, вычислимыми по Тьюрингу (без доказательства).

Неприменимость машины Тьюринга.
Пусть m машинное слово в алфавите $A \cup Q$ машины Т. Вычеркнем из слова m букву a_0 и буквы из Q. Получаем редуцированное слово [m].
Редуцированное слово [m] перерабатывается машиной Т в(редуцированное) слово b ,если для некоторой программы β и некоторго р/

$$q_1 a_0[m] \to (q_1 a_0[m])^{(1)} \to \ldots \to (q_1 a_o[m])^{(p)}$$
$$q_0 \in (q_1 a_0[m])^{(p)}, \quad b = [(q_1 a_0[m])^{(p)}]$$

Символически факт переработки слова [m] записываем в виде

$$b = \beta([m])$$

Если ни для какого натурального p не выполняется условие

$q_0 \in (q_1 a_0[m])^{(p)}$

то говорят,что машина *неприменима* к слову [m]. В этом случае выражение $\beta([m])$ является неопределенным.

Тезис Тьюринга. Все вычислимые частичные функции вычисляются на машинах Тьюринга-Поста.

Теорема 1 Все частично словарные функции,вычислимые на машине Тьюринга,являются частично рекурсивными.

Теорема 2 Для любой частично рекурсивной функции существует вычисляющая ее машина Тьюринга.

Билет 36 . Определение нормального алгоритма Маркова и порядок его работы.

Определение. Нормальный алгоритм Маркова задается алфавитом A в котором он работает и списком подстановок. Список подстановок это функциональная схема алгоритма. Так же как МТ каждый НАМ предназначен для решения задач определенной массовой проблемы.

Порядок работы.

Порядок работы НАМ N над словом $\alpha_0 \in A^*$ состоит из выполнения одним за другим однотипных шагов. На шаге i=0 зарождается исходное слово на последующих шагах функциональная схема алгоритма применяется к α_{i-1} и перерабатывает его в α_i

$\alpha_{i-1} \mid \rightarrow \alpha_i$

Таким образом в процессе работы алгоритма получается последовательность слов: $\alpha_0, \alpha_1, \ldots \alpha i-1, \alpha_i, \ldots$

Процесс работы НАМ заканчивается после выполнения шага K=0,1,2,... На слове α_k , если на шаге k осуществимо действие заключительной перестановки или на шаге k+1 список подстановок не может переработать слово α_k

Билет 37. Пример работы нормального алгоритма Маркова. Тезис Маркова. Теорема об эквивалентности машин Тьюринга и нормальных алгоритмов Маркова.

Пример:

$$N: \begin{cases} 101 \Rightarrow 1 \\ 01 \rightarrow 1 \\ 0 \rightarrow 00 \end{cases}$$

a) $\alpha_0 = 11$ - работа алгоритма заканчивается после шага 0, т.к. алгоритм не может переработать α_0.

b) $\alpha_0 = 001$

$0\underline{01}$

$\underline{01}1$ - результат работы $N(001) = 111$ (111 переработать не может)

111

c) $\alpha_0 = 10101$

$\underline{101}01$

101 - $N(10101) = 101$ (т.к. 101 – заключительная подстановка)

d) $\alpha_0 = 10$

$1\underline{0}$

$1\underline{00}$

$1\underline{000}$ - алгоритм не применим, т.к. бесконечен.

.....

Тезис Маркова:

Любой алгоритм в алфавите A может быть реализован некоторым нормальным алгоритмом над алфавитом A.

Теорема об эквивалентности МТ и НАМ:

Какова бы ни была МТ – T (НАМ - N) в алфавите A, существует НАМ N (МТ T) над алфавитом A такой, что для всех слов $\alpha, \beta \in A^*$, справедливо:

$\beta = N(\alpha)$ тогда и только тогда, когда $\beta = T(\alpha)$

Билет 38. Сравнительный анализ трех типов алгоритмических моделей. Оценка сложности алгоритма.

Анализ трех типов моделей показал, что основные свойства алгоритмов дискретности, детерминированности, массовости и результативности остаются неизменными для различных способов описания. Все три алгоритмических модели эквивалентны, что позволяет рассматривать вычислительный алгоритм инвариантным к способу описания.

Различия наблюдаются в использовании конструктивных объектов.

	Рекурсивные функции	**Машины Тьюринга**	**Нормальные алгоритмы Маркова**
Конструктивные объекты	Числовые функции, определенные на множестве натуральных чисел с нулем.	Символы алфавитов внешней памяти (на информационной ленте) и внутренней памяти (состояний управляющего устройства).	Слова.
Задание процесса вычисления	Операторами суперпозиции, примитивной рекурсии и минимизации.	Протоколом, использующим функции перемещения управляющего устройства, а также функции выхода (записи нового символа на ленте) и перехода (из одного внутреннего состояния в другое).	Правилами подстановки, изменяющими состав и структуру исходного слова до искомого результата.

Все алгоритмические модели обеспечивают способ получения значений вычислимой функции, причем, согласно тезису Черча, совпадают множества вычислимых и частично рекурсивных функций, а также множества всюду определенных вычислимых и общерекурсивных функций.

При реализации алгоритма с привлечением одной из трех моделей возникает задача **оценки сложности алгоритма**. Выделяют ***сложность описания*** алгоритма и ***сложность вычисления*** алгоритма.

Сложность описания алгоритма есть величина, характеризующая длину описания алгоритма.

	Рекурсивные функции	**Машины Тьюринга**	**Нормальные алгоритмы Маркова**
Сложность описания алгоритма	Число букв и символов, используемых в описании операторов.	Число команд.	Число правил подстановки.

Сложность вычисления алгоритма есть функция, дающая числовую оценку трудоемкости применения алгоритма к исходным данным для получения искомого результата. Чаще всего рассматриваются время и объем памяти, необходимые для вычисления.

Время вычисления алгоритма характеризуется произведением числа шагов алгоритма от исходных данных до искомого результата на среднее физическое время реализации одного шага алгоритма. Число шагов алгоритма определяется его описанием в данной алгоритмической модели.

Среднее физическое время зависит от типа компьютера, способов хранения и выборки данных и скорости обработки информации.

Объем памяти, как количественная характеристика алгоритма, определяется количеством единиц памяти, используемых в процессе вычисления алгоритма. Эта величина не может превосходить максимального числа единиц памяти, используемых в данном компьютере на одном шаге алгоритма.

Основной задачей программиста является разработка эффективного алгоритма, который позволял бы достичь требуемого результата при минимальных затратах времени и памяти.

Билет 39. Алгоритмически неразрешимые проблемы: проблема остановки машины Тьюринга, проблема ее самоприменимости, проблема эквивалентности слов в ассоциативном исчислении.

*Рассмотрим следующую задачу***.** По любому алгоритму *A* и данным *a* определить, приведет ли к результату работа *A* при исходных данных *a*. Иначе говоря, нужно построить алгоритм *B* такой, что *B*(*A, a*)=*И*, если *A*(*a*) дает результат, и *B*(*A*, *a*)=*Л*, если *A*(*a*) не дает результата. В силу тезиса Тьюринга эту задачу можно сформулировать как задачу о построении машины Тьюринга: построить машину T_0, такую, что для любой машины Тьюринга T и любых исходных данных *a* для машины T $T_0(S_T, a)$=И, если машина T(*a*) останавливается, и $T_0(S_T, a)$=Л, если машина T(*a*) не останавливается. (Здесь S_T – система команд машины T). Эта задача называется проблемой остановки машины Тьюринга.

Теорема о неразрешимости проблемы остановки для произвольной машины Тьюринга. Не существует машины Тьюринга T_0, решающей проблему остановки для произвольной машины Тьюринга Т.
В силу тезиса Тьюринга невозможность построения машины Тьюринга означает отсутствие алгоритма решения данной проблемы. Поэтому полученная теорема дает первый пример *алгоритмически неразрешимой проблемы* (и потому у нее такое название!).
Важное замечание. В утверждениях об алгоритмической неразрешимости речь идет об отсутствии единого алгоритма, решающего данную проблему; при этом вовсе не исключается возможность решения этой проблемы в частных случаях, но различными средствами для каждого случая. В частности, данная теорема не исключает того, что для отдельных классов машин Тьюринга проблема остановки может быть решена. Поэтому неразрешимость общей проблемы остановки вовсе не снимает необходимости доказывать сходимость предлагаемых алгоритмов, а лишь показывает, что поиск таких доказательств нельзя полностью автоматизировать. Неразрешимость проблемы остановки можно интерпретировать как несуществование общего алгоритма для отладки программ, точнее, алгоритма, который по тексту любой программы и данным для нее определял бы, зациклится ли программа на этих данных или нет. Если учесть сделанное ранее замечание, такая интерпретация не противоречит тому эмпирическому факту, что большинство программ в конце концов удается отладить, т. е. установить наличие зацикливания, найти его причину и устранить ее. При этом решающую роль играют опыт и искусство программиста.
Частным случаем проблемы остановки является проблема самоприменимости. Суть этой проблемы заключается в следующем. Программу машины Тьюринга можно закодировать каким-либо определенным шифром. На ленте машины можно изобразить ее же собственный шифр, записанный в алфавите машины. Здесь, как и в случае обычной программы возможны два случая:

- машина применима к своему шифру, т.е. она перерабатывает этот шифр и после конечного числа тактов останавливается;
- машина неприменима к своему шифру, т. е. машина никогда не переходит в стоп-состояние.

Таким образом, сами машины (или их шифры) разбиваются на два класса: самоприменимых и несамоприменимых тьюринговых машин. Проблема заключается в следующем: как по любому заданному шифру установить, к какому классу относится машина, зашифрованная им, к классу самоприменимых или несамоприменимых.
Теорема. Проблема распознавания самоприменимости алгоритмически неразрешима.
Еще одной алгоритмически неразрешимой проблемой является проблема эквивалентности слов для ассоциативных исчислений.
Рассмотрим некоторый алфавит $A=\{a, b, c,...\}$ и множество слов в этом алфавите. Будем рассматривать преобразования одних слов в другие с помощью некоторых допустимых подстановок $\alpha \to \beta$, где α и β — два слова в том же алфавите А. Если слово *y* содержит α как подслово, например, $\alpha_1\alpha\alpha_2\alpha_3\alpha$, то возможны следующие подстановки: $\alpha_1\beta\alpha_2\alpha_3\alpha$, $\alpha_1\alpha\alpha_2\alpha_3\beta$, $\alpha_1\beta\alpha_2\alpha_3\beta$.
Ассоциативным исчислением называется совокупность всех слов в некотором алфавите вместе с какой-нибудь конечной системой допустимых подстановок. Для задания ассоциативного исчисления достаточно задать соответствующий алфавит и систему подстановок. Очевидно, что одним из примеров ассоциативного исчисления являются нормальные алгоритмы Маркова.
Если слово *R* может быть преобразовано в слово *S* путем однократного применения определенной подстановки, то *R* и *S* называются смежными словами. Последовательность слов $R_1, R_2, \ldots, R_{n-1}, R_n$ таких, что все пары слов $(R_i, R_{i+1}), i=1,2,..., n-1$ являются смежными, называется *дедуктивной цепочкой,* ведущей от слова R_1 к слову R_n. Если существует цепочка, ведущая от слова *R* к слову *S*, то *R* и *S* называются эквивалентными: $R \sim S$.
Для каждого ассоциативного исчисления возникает своя специальная проблема эквивалентности слов: для любых двух слов в данном исчислении требуется узнать, эквивалентны они или нет.
Теорема. Проблема эквивалентности слов в любом ассоциативном исчислении алгоритмически неразрешима.
Эта проблема решена лишь в некоторых ассоциативных исчислениях специального вида.

Билет 40. Проблема соответствий Поста. Теорема Райса (без доказательства) и ее смысл.

Еще одной алгоритмически неразрешимой проблемой является проблема соответствий Поста.

Определение 1. *Постовской системой соответствия* над алфавитом A называется упорядоченная пара конечных последовательностей $((x_1,..,x_n),(y_1,...,y_n))$, где $x_i \in A^*$ и $y_i \in A^*$ для всех i.

Замечание. Систему $((x_1,..,x_n),(y_1,...,y_n))$ иногда изображают в виде $\left[\frac{x_1}{y_1}\right],...,\left[\frac{x_n}{y_n}\right]$.

Определение 2. *Решением* постовской системы соответствия $((x_1,..,x_n),(y_1,...,y_n))$ называется непустая последовательность индексов $(i_1,..,i_k)$, удовлетворяющая условию $x_{i_1}...x_{i_k} = y_{i_1}...y_{i_k}$, где $1 \le i_j \le n$ для каждого j.

Пример. Пусть $A=\{a, b, c\}$. Рассмотрим постовскую систему соответствия $\left[\frac{aab}{a}\right],\left[\frac{a}{aa}\right],\left[\frac{caa}{bc}\right]$.

Последовательность $(2,1,3,2,2)$ является решением этой системы, так как
$a \cdot aab \cdot caa \cdot a \cdot a = aa \cdot a \cdot bc \cdot aa \cdot aa$.

Определение 3. *Проблемой соответствий Поста* (комбинаторная проблема Поста, Post correspondence problem) называется проблема нахождения алгоритма, выясняющего для каждой постовскои системы соответствия, существует ли решение этой системы.

Теорема. Пусть $|A|\ge 2$. Тогда не существует алгоритма, позволяющего по произвольной постовской системе соответствия над алфавитом A узнать, имеет ли она решение. (Другими словами, проблема соответствий Поста неразрешима.)

Неразрешимость данной проблемы широко используется для доказательства неразрешимости других алгоритмических проблем.

Напомним, что еще одной, ранее рассмотренной в данном курсе алгоритмически неразрешимой проблемой, является проблема разрешимости формул логики предикатов.

Теорема Райса. Никакое нетривиальное свойство вычислимых функций не является алгоритмически разрешимым.

Смысл этой теоремы заключается в том, что по описанию алгоритма ничего нельзя узнать о свойствах функции (является ли она периодической, ограниченной, монотонной, содержит ли среди своих значений заданное число и т.п.), которую он вычисляет. В частности, оказывается неразрешимой проблема эквивалентности алгоритмов: по двум заданным алгоритмам нельзя узнать, вычисляют они одну и ту же функцию или нет.

Опытного программиста теорема Райса не должна удивлять: он знает, что по тексту сколь-нибудь сложной программы, не запуская ее в работу, трудно понять, какую функцию она вычисляет, не имея гипотез о том, что она должна делать.

Замечание. Использованное здесь выражение “ничего нельзя узнать” является, вообще говоря, преувеличением. Его точный эквивалент – “не существует единого алгоритма, позволяющего узнать”. К тому же речь идет о невозможности распознавания свойств вычислимых функций, записанных на универсальном алгоритмическом языке (языке рекурсивных функций, машин Тьюринга или нормальных алгоритмов Маркова). Свойства подклассов вычислимых функций, описанных в специальных языках, вполне могут оказаться разрешимыми.

Билет 41. Особенности прикладных исчислений. Аксиомы для равенства.

ИП не содержащие функциональных букв и предметных констант называются частными ИП. Прикладные ИП характеризуются тем, что в них добавляются собственные аксиомы, содержащие индивидуальные функциональные буквы и предметные константы.

Кроме того в системах аксиом (P1) и (P2) участвуют уже не предметные константы а произвольные …

(P1') $\forall x\, F(x) \to F(t)$,

(P2') $F(t) \to \exists x\, F(x)$,

Большинство прикладных исчислений содержит предикат равенства = и определяющие его аксиомы. Аксиомами для равенства могут служить следующие.

(E1) $\forall x\, x = x$ (конкретная аксиома),

(E2) $(x = y) \to (F(x,x) \to F(x,y))$ (схема аксиом),

где $F(x,y)$ – получается из $F(x,x)$ заменой некоторых (не обязательно всех) вхождений x на y при условии, что y в этих вхождениях также остается свободным. Всякая теория, в которой (E1) и (E2) являются теоремами или аксиомами, называется *теорией* (или исчислением) *с равенством.* Дело в том, что из (E1) и (E2) выводимы **основные свойства равенства** - рефлексивность, симметричность и транзитивность.

Теорема. В любой теории с равенством:

1) $\vdash t=t$ для любого терма t;

2) $\vdash x = y \to y = x$;

3) $\vdash x = y \to (y = z \to x = z)$.

Аксиомы:

A1. $F(0) \,\&\, \forall x\,(F(x) \to F(x')) \to \forall x\, F(x)$ – принцип индукции,

A2. $t'_1 = t'_2 \to t_1 = t_2$,

A3. $\overline{t' = 0}$,

A4. $t_1 = t_2 \to (t_1 = t_3 \to t_2 = t_3)$,

A5. $t_1 = t_2 \to t'_1 = t'_2$,

A6. $t + 0 = t$,

A7. $t_1 + t'_2 = (t_1 + t_2)'$,

A8. $t \cdot 0 = 0$,

A9. $t_1 \cdot t'_2 = t_1 \cdot t_2 + t_1$.

Билет 42. Формальная арифметика. Теоремы Геделя о неполноте (без доказательства) и их смысл.

Наиболее изученной формальной теорией, которая играет фундаментальную роль в основаниях математики, является формальная арифметика – прикладное исчисление предикатов, в котором имеются:

1. Предметная константа 0.
2. Двухместные факторы сложения и умножения, одноместный фактор ‘.
3. Двухместный предикат =.
4. Собственные аксиомы (схемы аксиом):

A1. $F(0) \,\&\, \forall x\,(F(x) \to F(x')) \to \forall x\, F(x)$ – принцип индукции,

A2. $t'_1 = t'_2 \to t_1 = t_2$,

A3. $\overline{t' = 0}$,

A4. $t_1 = t_2 \to (t_1 = t_3 \to t_2 = t_3)$,

A5. $t_1 = t_2 \to t'_1 = t'_2$,

A6. $t + 0 = t$,

A7. $t_1 + t'_2 = (t_1 + t_2)'$,

A8. $t \cdot 0 = 0$,

A9. $t_1 \cdot t'_2 = t_1 \cdot t_2 + t_1$.

Первая теорема Геделя о неполноте:

Любая формальная теория T содержащая формальную арифметику неполна. В ней существует и может быть эффективно построена формула F, такая что $\overline{F}$ истинно, но ни F ни $\overline{F}$ не выводимы в теории T.

Вторая теорема Геделя о неполноте:

Для любой непротиворечивой формальной теории T, содержащей формальную арифметику, формула выражающая непротиворечивость T, не доказуема в теории T.

Теоремы без доказательств!