Введение в математическую логику

Лекция 2

С первой лекции:

Что такое доказуемое математическое утверждение?

• Исчисления. Интегральное исчисление. Школьное исчисление уравнений. Грамматики. Тезис Гильберта

Кодирования

Взаимно-однозначное соответствие между натуральными числами и словами – выписывание всех слов «по порядку»: короткие раньше длинных, нули раньше единиц, сначала меняем последние буквы, потом первые.

- $0 \leftrightarrow \Lambda$
- $1 \leftrightarrow 0$
- $2 \leftrightarrow 1$
- $3 \leftrightarrow 00$
- $4 \leftrightarrow 01$
- $5 \leftrightarrow 10$
- $6 \leftrightarrow 11$
- $7 \leftrightarrow 000$
- . . .

Кодирования

Взаимно-однозначное соответствие между натуральными числами и парами натуральных чисел – выписывание всех пар «по порядку»: по возрастанию суммы, при одной сумме уменьшаем второй компонент: $n= (a+b)(a+b+1)/2 + a$

- 0 $\leftrightarrow$ <0,0>
- 1 $\leftrightarrow$ <0,1>
- 2 $\leftrightarrow$ <1,0>
- 3 $\leftrightarrow$ <0,2>
- 4 $\leftrightarrow$ <1,1>
- 5 $\leftrightarrow$ <2,0>
- 6 $\leftrightarrow$ <0,3>
- 7 $\leftrightarrow$ <1,2>
- . . .

	0	**1**	**2**	**3**
0	0 <0,0>	1 <0,1>	3 <0,2>	6 <0,3>
1	2 <1,0>	4 <1,1>	7 <1,2>	11 <1,3>
2	5 <2,0>	8 <2,1>	12 <2,2>	17 <2,3>
3	9 <3,0>	13 <3,1>	18 <3,2>	24 <3,3>

Кодирования

Взаимно-однозначное соответствие между словами и парами слов:

- $\Lambda \leftrightarrow \langle\Lambda,\Lambda\rangle$
- $0 \leftrightarrow \langle\Lambda,0\rangle$
- $1 \leftrightarrow \langle 0,\Lambda\rangle$
- $00 \leftrightarrow \langle\Lambda,1\rangle$
- $01 \leftrightarrow \langle 0,0\rangle$
- $10 \leftrightarrow \langle 1,\Lambda\rangle$
- $11 \leftrightarrow \langle\Lambda,00\rangle$
- . . .

	Λ	**0**	**1**	**00**
Λ	Λ $\langle\Lambda,\Lambda\rangle$	0 $\langle\Lambda,0\rangle$	00 $\langle\Lambda,1\rangle$	11 $\langle\Lambda,00\rangle$
0	1 $\langle 0,\Lambda\rangle$	01 $\langle 0,0\rangle$	000 $\langle 0,1\rangle$	100 $\langle 0,00\rangle$
1	10 $\langle 1,\Lambda\rangle$	001 $\langle 1,0\rangle$	101 $\langle 1,1\rangle$	1010 $\langle 1,00\rangle$
00	010 $\langle 00,\Lambda\rangle$	110 $\langle 00,0\rangle$	1011 $\langle 00,1\rangle$	10001 $\langle 00,00\rangle$

Задача. Продолжить заполнение таблицы. Оценить длину кода пары слов через сумму длин элементов пары.

Что такое вычислимая математическая функция?

Определение. Вычислимая функция – это функция, для которой существует алгоритм, ее вычисляющий.

- Аргументы функции – слова в некотором алфавите. Числа – тоже слова, или кодируются словами. Бывают и другие конструктивные объекты.

Свойства алгоритма:

- Описание алгоритма (программа)
- Начальное состояние выполнения («вычисления»), получаемое из исходного данного
- Раздельные шаги перехода к однозначно определяемому следующему состоянию, или остановка
- Из состояния, в котором алгоритм остановился, получается («печатается») значение функции

Исходные данные: слова в некотором алфавите. У нас – двоичные слова (алфавит {0,1}).

Состояние выполнения (вычисления) – конструктивный объект (записывается словом). Описание алгоритма – конструктивный объект (записывается словом).

Область определения вычислимой функции

- На некотором исходном данном алгоритм может не останавливаться. Он *не применим* к этому данному.

  То есть функция может быть определена не на всех словах (частичная функция).

- Можно ли продолжить функцию до всюду определенной? Можно ли продолжить ее до вычислимой всюду определенной?

Функции нескольких аргументов

- Набор (вектор) аргументов $< x_0 , x_1 , x_2 ,\ldots x_n >$ часто будем обозначать одной буквой *x.*
- Наборы слов фиксированной длины также могут кодироваться словами.
- Все конечные последовательности слов в любом фиксированном алфавите можно одновременно закодировать словами. (Алфавит может быть и счетным.)

Теорема. Суперпозиция (композиция) вычислимых функций вычислима.

- Верно и для функций многих аргументов.

Перечислимые множества

Определение. Перечислимое множество – это множество значений вычислимой функции.

Теорема. Следующие утверждения о множестве эквивалентны:

1. Множество перечислимо.

2. Множество пусто или является множеством значений всюду определенной вычислимой функции.

3. Множество является областью определения вычислимой функции.

Доказательство. Для пустого множества все верно. Пусть слово *c* – из множества.

$1 \Rightarrow 2$

Входное слово - код пары <аргумент, число шагов> = <*x*,*t*>. Если на *x* за время *t* (или раньше) вычисление, моделирующее (1), дало результат, то выдаем его, если не дало, то выдаем *c*.

Перечислимые множества

Теорема. Следующие утверждения о множестве эквивалентны:

1. Множество перечислимо.

2. Множество пусто или является множеством значений всюду определенной вычислимой функции.

3. Множество является областью определения вычислимой функции.

Доказательство.

$2 \Rightarrow 3$

Получив входное слово, начинаем вычислять функцию (2) последовательно на всех словах, если входное слово окажется значением, то выдаем c.

$3 \Rightarrow 1$

Получив входное слово, вычисляем функцию (3). Если процесс закончится, выдаем входное слово, как результат.

Перечислимые множества

Теорема. Следующие утверждения о множестве эквивалентны:

1. Множество перечислимо.

2. Множество породимо.

Доказательство.

1 (область определения) $\Rightarrow$ 2

Вывод состоит в том, чтобы выписать произвольное слово, а потом пытаться применить к нему функцию. Если функция применится, то слово – выведено.

2 $\Rightarrow$ 1

Возьмем грамматику, порождающую данное множество (тезис Гильберта). Получив входное слово, рассматриваем его как код цепочки слов. Проверяем, является ли цепочка выводом в данной грамматике. Если да, то результат вывода выдаем как результат вычисления функции.

Операции над перечислимыми множествами

Теорема. Объединение перечислимых множеств перечислимо.

Доказательство. Пусть два множества заданы, как области определения двух функций. Запускаем два алгоритма параллельно. Когда один из них закончит работу – останавливаемся.

Теорема. Пересечение перечислимых множеств перечислимо.

Доказательство. Пусть два множества заданы, как области определения двух функций. Запускаем два алгоритма параллельно. Когда оба закончат работу – останавливаемся.

Теорема. График вычислимой функции – перечислим.

Доказательство. Вычисляем функцию, к результату присоединяем аргумент.

Теорема? Дополнение к перечислимому множеству перечислимо?

- Как установить, что слово
  НЕ является значением функции?
- Как узнать, что утверждение –
  НЕ доказуемо?

Разрешимость

Определение. Характеристическая функция данного множества S – всюду определенная функция, равная 1 на S и нулю – вне S.

Определение. Множество разрешимо, если его характеристическая функция вычислима.

Теорема. Объединение, пересечение и дополнение (!) разрешимых множеств разрешимы.

Доказательство. Вычисление характеристических функций.

Теорема. Множество разрешимо титтк оно и его дополнение перечислимы.

Доказательство. Разрешимость $\Rightarrow$ Перечислимость (область определения) – бесконечная работа на 0.

Перечислимость множества и дополнения (область определения) $\Rightarrow$ Разрешимость – запускаем одновременно вычисление функций для которых ОО есть множество и его дополнение. Одна из функций будет вычислена, и это дает ответ на вопрос о принадлежности.

Алгоритмические проблемы

- Проблемы построения алгоритмов
- Проблема построения алгоритма разрешения

10-ая Проблема Гильберта

- Построить алгоритм, который по всякому алгебраическому уравнению от нескольких неизвестных с целыми коэффициентами, выясняет, есть ли у него решение в целых числах.

Алгоритмы Маркова («нормальные алгорифмы»)

Определение. Алгоритм Маркова Φ – это набор < Σ, Π >, где:

- Σ – *алфавит* Φ, у нас {0,1},
- Π – набор слов вида *u* $\to$ *v*, *u* $\to \bullet$ *v* (*правил* Φ), слова *u*, *v* – в алфавите Δ, включающем Σ, но $\to$, $\bullet \notin \Delta$,
- *u* называется левой частью правила, *v* – правой,
- правила, содержащие $\to\bullet$, называются заключительными.

Применение алгоритма к слову w в алфавите Δ:

Построение последовательности слов, начинающейся с *w* и такой, что следующее слово получается из предыдущего слова *s* действием:

- Выбрать в Π первое правило, для которого левая часть входит в *s.*
- Просматривая слово s слева направо, заменить левую часть правила на правую, как только это окажется возможным.

Если правила не удалось найти, или выбранное правило – заключительное, то построение заканчивается и последнее построенное слово – результат применения Φ к слову *w*. В противном случае построение продолжается.

Если последовательность не заканчивается (оказывается бесконечной), то говорят, что алгоритм Φ *не применим* к слову *w*.

Пример:

Программа:

1. |0 → 0||
2. 1 → 0|
3. 0 →

Исходное данное 101

- 0|01
- 00||1
- 00||0|
- 00|0|||
- 000|||||
- 00|||||
- 0|||||
- |||||

Тезис Чёрча

- Всякая вычислимая функция вычислима некоторым алгоритмом Маркова.