Коллоквиум по дискретной математике №1.

oleander_twig

Содержание

1 Формат проведения.

Студент получает билет, содержащий по одному вопросу из каждой темы, и имеет не менее 45-ти минут на подготовку ответа. При подготовке разрешается пользоваться любыми материалами, но не советоваться с кем-либо. Во время ответа можно опираться на предварительно написанный текст, но принимающий может попросить изменить порядок изложения или предложить ответить на какие-либо вопросы «с чистого листа».
Все утверждения, кроме особо отмеченных, нужно уметь доказывать. Излагая определения и формулировки, надо уметь приводить соответствующие примеры и контрпримеры. Принимающий вправе задавать дополнительно любые разумные вопросы в рамках программы, но *обычно* вне пределов билета не спрашивает доказательств утверждений.
За ответ на каждый из вопросов билета ставится от 0 до 3-х баллов. Дополнительные вопросы *могут* задаваться при наличии существенных недостатков в ответе на вопросы билеты. Ответы на них *могут* улучшить оценку за ответ на вопросы билета.

2 Вычислимость.

2.1 Вычислимые функции (при интуитивном понимании алгоритма). Разрешимые и перечислимые множества. Связь конечности, разрешимости и перечислимости. Разрешимые множества под действием операций алгебры множеств и декартова произведения.

Опр. Функция $f\colon A \xrightarrow{p} B$ **вычислима**, если есть алгоритм, который её вычисляет.

Алгоритм вычисляет функцию $f\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$, если $\forall x \in \mathbb{N}$:
1) на любом входе $x \in dom\, f$ алгоритм вычисляет $f(x)$ и завершается;
2) и на любом входе $x \in A \setminus dom\ f$ алгоритм не завершается ни за какое конечное количество шагов.

Пример.

$$f(x) = \begin{cases} 1, if\ God\ exists \\ 0, if\ doesnot \end{cases}$$

Псевдокод, вычисляющий эту функцию может выглядеть по-разному:

```
int f( int x ) {
return 1;
}
```

```
int f( int x ) {
return 0;
}
```

Но несмотря на то, что мы не можем предъявить алгоритм я явном виде, мы все еще можем утверждать, что функция f вычислима, так как нам достаточно всего лишь доказать существование алгоритма, вычисляющего нашего функцию.

Опр. Множество $A \subseteq \mathbb{N}$ **разрешимо**, если его характеристическая функция

$$\chi_A(x) = \begin{cases} 1, if\ x \in A \\ 0, if\ x \notin A \end{cases}$$

вычислима. Иными словами, если есть программа, по входу определяющая, принадлежит ли этот вход множеству A; выдающая 1, если принадлежит, и 0 в противном случае.

Опр. Множество $A \subseteq \mathbb{N}$ **перечислимо**, если есть программа, на пустом входе последовательно выписывающая все элементы A *и только их*. Уточним это понятие. Можно заметить, что выписывание каждого элемента может занимать несколько шагов. Важно потребовать, чтобы промежуточные записи не считались выписанными элементами — для этого достаточно сказать, что элемент выписан, если он написан на ленте и программа находится в некотором выделенном состоянии. Тогда перечислимость означает, что для каждого элемента A найдется шаг исполнения программы, когда этот элемент написан на ленте (возможно, вместе с другими) и программа в выделенном состоянии.

Утверждение. Если A - конечно, то A - разрешимо.

Доказательство. Составим разрешающую процедуру (ну или предъявим характеристическую функцию). Пусть $A = \{a_1, a_2, ..., a_n\}$.

```
int in_A( int x ) {
return (x == a_{1}) || (x ==a_{2}) || ... || (x ==a_{n})
}
```

Если A конечно, то мы все иксы сможем запихнуть в программу, а значит такое алгоритмическое описание имеет место быть. ∎

Утверждение. Если A и B разрешимы, то разрешимы $\overline{A}, A \cup B, A \cap B, A \times B$.

Доказательство.

$$\chi_{A\cap B}(n) = \chi_A(n) \cdot \chi_B(n)$$

$$\chi_{A\cup B}(n) = \chi_A(n) + \chi_B(n) - \chi_A(n) \cdot \chi_B(n)$$

$$\chi_{\overline{A}}(n) = 1 - \chi_A(n)$$

$$\chi_{A\times B}(n, m) = \chi_A(n) \cdot \chi_B(m)$$

∎

Утверждение. Если A разрешимо, то A перечислимо.

Доказательство. Нам нужен алгоритм, который будет работать бесконечно долго и печатать элементы из A, при условии что оно разрешимо.

```
int n = 0;
while (1) {
if (in_A(n)) {print (n);}
n++;
}
```

Натуральные числа мы перечислить можем. Понятно, что последовательно просматривая каждое число из $\mathbb{N}$, мы напечатаем все числа из A. ■

Заметим! Из конечности следует разрешимость. Из разрешимости следует перечислимость. (В обратную сторону не работает).

2.2 Перечислимые множества под действием операций алгебры множеств, декартова произведения и проекции. Теорема Поста.

Теорема Поста. Множество A - разрешимо $\Longleftrightarrow$ A и $\overline{A}$ - перечислимы.

Доказательство. $\Rightarrow$ Мы знаем, что из разрешимости множества A следует перечислимость множества A; если разрешимо множество A, то множество $\overline{A}$ тоже, а отсюда следует перечислимоть множества $\overline{A}$.
$\Leftarrow$ У нас A и $\overline{A}$ перечислимы, а значит есть алгоритмы, которые их перечилсяют, так называемые "перечислители".
Пусть $\tilde{A}$ ("A краисвое") перечисляет A и $\tilde{B}$ перечисляет $\overline{A}$. Поочередно будем делать по шагу алгоритмов $\tilde{A}$ и $\tilde{B}$.
1) Если на каком-то шаге $\tilde{A}$ вывел n, то $n \in A$(учитывая ,что $\tilde{A}$ перечисляет множество A), тогда мы выводим 1 и останавливаемся.
2) Если на каком-то шаге $\tilde{B}$ вывел n, то $n \in \overline{A}$, тогда мы выводим 0 и останавливаемся.
Заметим, что $A \cup \overline{A} = \mathbb{N}$. Поэтому одно из событий обязательно случится $\Rightarrow$ наш алгоритм завершится. (Заострю внимание на том, что если функция определена в точке x, то алгоритм обязательно должен завершаться в этой точке. Это мы и доказали.) ■

Пусть A - подмножество наборов длины k натуральных чисел, $A \subseteq \mathbb{N}^k$. Тогда **проекцией** на i-ую координату называется множество всевозможных i-ых координат наоборов из A.

$$i \in \{1, ..., k\}$$
$$pr^i\ A = \{x \mid \exists x_1 ... \exists x_{i-1} \exists x_{i+1} ... \exists x_k (x_1, ..., x_{i-1}, x, x_{i+1}, ..., x_k) \in A\}$$

Утверждение. Если A и B перечислимы, то перечислимы $A \cup B, A \cap B, A \times B, pr^i A$.

Доказательство. Пусть $\tilde{A}$ ("А краисвое") перечисляет А и $\tilde{B}$ перечисляет В.
Запустим $\tilde{A}$ и для каждого выведенного набора $(a_1, ..., a_k)$ будем печатать его i-ую координату. Понятно, что таким образом мы перечислим все i-ые координаты множества А. [Доказали для проекции.]
Будем поочередно делать шаги перечислетелей $\tilde{A}$ и $\tilde{B}$. Все выведенные кем-то из них числа отправляем в выходной поток. (Добавлю от себя. Так как А и Б перечислимы, то найдётся шаг на котором выводится a_i и b_i соответственно. Идя поочередно на двух перечислителях, мы гарантированно пройдем все шаги вывода элементов.) [Доказали для объединения.]
Будем поочередно делать шаги перечислетелей $\tilde{A}$ и $\tilde{B}$. Всё, что они выпишут, мы будем накапливать в некоторых буферах $A^{'}$ и $B^{'}$. Пусть $A_i^{'}$ - то, что лежит в буфере $A^{'}$ после i-го шага $\tilde{A}$. Аналогично для $\tilde{B}$.
После i-го шага $\tilde{A}$ и i-го шага $\tilde{B}$, выводим множество $A_i^{'} \cap B_i^{'}$. Заметим, что множества $A_i^{'}$ и $B_i^{'}$ конечные, а значит их пересечение тоже. Тогда мы легко можем вывести элементы множества $A_i^{'} \cap B_i^{'}$ (перебираем элементы из А и сравниваем с элементами из В). Причем, мы это сделаем за конечное время $\Longrightarrow$ сможем перейти к $i+1$ шагу перечислителей. [Доказали для пересечения.]
Для декартового произведения мы проделываем всё тоже самое, только на i-ом шаге выводим $A_i^{'} \times B_i^{'}$. Это конечная вещь. ■

2.3 Теорема о графике вычислимой функции. Перечислимость образа и прообраза множества под действием вычислимой функции.

$f\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$
График функции f: $\Gamma_f = \{(x, y) \in \mathbb{N}^2 \mid f(x) = y\}$.
С точки зрения теории множеств график функции и сама функции ничем не отличаются. Но нам удобней разделить для некоторых штук. Например, для следующей теоремы.

Теорема о графике вычислимой функции. Пусть $f : \mathbb{N} \xrightarrow{p} \mathbb{N}$, тогда f - вычислима $\Longleftrightarrow \Gamma_f$ - перечислим.

Доказательство. $\Longleftarrow$ Мы хотим вычислить функцию, то есть у нас есть x и мы хотим получиться $f(x)$, если оно существует.
Запускаем перечислитель графика Γ_f и ждем первой пары вида (x, y). $(x, y) \in \Gamma_f \Longrightarrow y = f(x)$. Дождавшись пары, выдаем y и завершаемся. Если же такой пары нет в графике, то алгоритм будет ждать её на вход вечно, то есть зациклится - как раз по определению.
$\Longrightarrow$ Хотим перечислить график Γ_f, то есть перечислить пары натуральных чисел.
Запускаю перечислитель всех троек натуральных чисел. Для каждой выведенной тройки (x, y, k) делаем k шагов алгоритма вычисления функции f на входе x. Если он

вывел y и остановился, то $f(x) = y \Longrightarrow$ печатаем пару (x, y), переходим к следующей тройке. ∎

Корректность выше упомянутого алгоритма.
Допустим$(x, y) \in \Gamma_f$. Тогда $f(x) = y \Longrightarrow \exists$ число k такое что алгоритм, вычисляющий функцию f, на входе x выдаст y за k шагов (по определению). Поэтому любая пара из графика будет выведена при рассмотрениии всех возможных троек (x, y, k).

Введём понятие **образа** множества A под действием вычислимой функции f. Обозначение: f(A). Образом является проекция графика на ось Ординат.

$$f(A) = pr^2(\Gamma_f \cap (A \times \mathbb{N}))$$

Следствие 1. Если f - вычислима и A перечислимо, то f(A) - перечислимо.

Доказательство. f вычислима $\Longrightarrow$ по теореме о графике Γ_f - перечислим. A - перечислимо $\Longrightarrow$ $A \times \mathbb{N}-$ перечислимо. Ну а пересечение двух перечислимых множеств тоже перечислимо $\Longrightarrow$ его проекция тоже. ∎

Аналогично введём понятие **прообраза** $f^{-1}(A)$.

$$f^{-1}(A) = pr^2(\Gamma_f \cap (\mathbb{N} \times A))$$

Следствие 1 для него тоже выполнено.

Следствие 2. Если f - вычислима, то $dom\ f$ и $rng\ f$ перечислимы.

Доказательство. $dom\ f = f^{-1}(\mathbb{N})$; $rng\ f = f(\mathbb{N})$ ∎

[Кажется, рассказывать это в билете не нужно, но пусть будет.]

2.4 Перечислимые множества суть, в точности, области определения вычислимых функций.

Опр. Если $A \subseteq \mathbb{N}$, то

$$\omega_A(n) \simeq \begin{cases} 1,\ if\ n \in A \\ undefined,\ if\ n \notin A \end{cases}$$

называется **полухарактерестической** функцией на множестве A.
Замечание. $dom\ \omega_A = A$.

Утверждение. Если A - перечислимо, то ω_A - вычислима.

Доказательство. Запустим перечислитель $\tilde{A}$ и ждём появления числа n (пусть оно поступило на вход). Когда число n появится, то выводим 1 и останавливаемся. ∎

Обратное тоже верно. Если ω_A—вычислима, то A - перечислимо.

Доказательство. $A = dom\ \omega_A$. А выше мы доказали, что область определения любой вычислимой функции - перечислима. ■

Следствие. A - перечислимо $\Longleftrightarrow$ ω_A— вычислима.

Утверждение. Если A - перечислимо и $A \neq \varnothing$, то $\exists$ вычислимая тотальная функция $f\colon \mathbb{N} \to \mathbb{N}$ такая что $A = rng\ f$ (т. е. $A = \{f(0), f(1), ..\}$).

Доказательство. У A есть перечислитель $\tilde{A}$. Так как $A \neq \varnothing$, то $\tilde{A}$ напечатает какое-то число $a \in A$ впервые на шаге k. А дальше мы заведём такую функцию:
$f(0) = a$
$f(n+1) =$ последнее число, которое $\tilde{A}$ напечатает за n + 1 + k шагов.
Не гарантируется, что за эти шаги будет что-то напечатано, но возможно, что-то будет. Во всяком случае, значения от f(0) до f(n+1) мы все можем посчитать благодаря нашему алгоритму. Поэтому, очевидно, что функция f - вычислима.
А еще очевидно, что она тотальна. Даже если наш перечислитель перестанет что-то печатать с какого-то шага, он будет повторять последний выведенный элемент. ■

Следствие. Если A - перечислимо, то $\exists f\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$ такая что $A = rng\ f$.

Утверждение. Если $A \subseteq \mathbb{N}$ - перечислимое множество, то $\exists B \subseteq \mathbb{N}^2$ - разрешимое множество пар, такое что $A = pr^i B$.

Доказательство. Пусть $\tilde{A}$ - перечислитель A; $B = \{(n,k) \in \mathbb{N}^2 \mid$ алгоритм выведет n на шаге k $\}$. Почему это множество разрешимо? - Если нам на вход дали пару (n, k), то мы можем по свойствам алгоритмов сделать k шагов и посмотреть вывел ли алгоритм $\tilde{A}$ n. Если да - множество разрешимо.
$pr^i B = \{n \in \mathbb{N} \mid \exists k$ что алгоритм на шаге k вывел n $\} \Longrightarrow \{n \in \mathbb{N} \mid n \in A\} = A$. ■

Обратное тоже верно. Если множество пар - разрешимо, то оно перечислимо. А как мы знаем, проекция перечислимого множества - перечислима.

Теорема о равносильных определениях перечислимых множеств. $\forall A \subseteq \mathbb{N}$ следующие утверждения равносильны.
1) A - перечислимо;
2) A - полуразрешимо (множество **полуразрешимо**, когда его полухарактеристическая функция вычислима);
3) $\exists f\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$ - вычислимая, такая что $A = dom\ f$;
4) $\exists f\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$ - вычислимая, такая что $A = rng\ f$;
5) $A = \varnothing$ или $\exists f\colon \mathbb{N} \to \mathbb{N}$ - вычислимая тотальная функция, такая что $A = rng\ f$;
6) $\exists B \subseteq \mathbb{N}^2$ - разрешимое множество B пар натуральных чисел, такое что $A = pr^i B$.

2.5 Непустые перечислимые множества суть, в точности, области значений вычислимых тотальных функций.

Я думаю, нужно сформулировать Теорему о равносильных определениях перечислимых множеств, но доказать только пятый пункт. Все необходимые для этого утверждения также содержаться в пункте выше.

2.6 Перечислимые множества суть, в точности, проекции разрешимых.

Я думаю, нужно сформулировать Теорему о равносильных определениях перечислимых множеств, но доказать только шестой пункт. Все необходимые для этого утверждения также содержаться в пункте 2.4.

2.7 Универсальная вычислимая функция (в классе вычислимых функций $\mathbf{N} \to \mathbf{N}$). Т-Предикат.