Введение в математическую логику

Лекция 13

Сложность. Подход теории алгоритмов

Сложность объекта

- да, да, да,… да (1 млн. раз)
- На экране нет миллиона «да».
- Есть описание объекта.
- Сложность объекта – минимальная длина его описания.

Сложность объекта

- Что такое описание?
- Аргумент для
  - машины,
  - алгоритма,
  - вычислимой функции,

  дающей объект.
- Вычислимая функция (дающая объект по его описанию) – способ описания.

Будем считать, что объекты и описания – двоичные слова.

- *Сложность при данном способе описания*

$$K_f(x) = \min \{|y| \mid f(y) = x\},$$

где $|y|$ – длина слова y;

если y нет, то $\min = \infty$.

Сложность объекта

- Описания бывают разные.
- f(0) = «сложный» объект (один).
- Есть ли способ описания, дающий самые короткие описания?
- Нет (очевидно).
- Можно ли получать описания, самые короткие «с точностью до» дополнительного слагаемого?
- Точная формулировка:
- **Теорема Колмогорова**. Существует способ описания u, такой, что для любого способа описания f найдется такое число C, что для всякого объекта x выполнено:

$$K_u(x) \leq K_f(x) + C .$$

Теорема Колмогорова

$$K_u(x) \le K_f(x) + C$$

- Д. Фиксируем некоторый вариант задания вычислимых функций алгоритмами, например, алгоритмами Маркова.
- Как мы видели на второй лекции, существует универсальная функция u:
для всякой вычислимой функции f, если p – задание (программа) f, то для всех y: f(y) = u(<p,y>).
- Возьмем произвольное x и такое y, что f(y) = x.

  $K_u(x) \le |\langle p,y\rangle|$, y можно взять самым коротким.
- Осталось доказать, что $|\langle p,y\rangle| \le |y| + C$.
- Здесь C может зависеть от p, но не от y.
- Каким нужно взять кодирование пар?

Кодирование пар

- Уже было на 2-ой лекции.
- Нужно добиться экономии по второму аргументу.
- Просто py?
- Где заканчивается p?
- Можно удвоить каждый символ p, а после p поставить 01.
- Тогда $|\langle p,y\rangle| = 2+2|p|+|y|$.
- Цель достигнута: $C = 2+2|p|$.
- Теорема Колмогорова доказана.
- Можно ли короче (достаточно знать длину p)?
- Почему нельзя короче, чем $|p|+|y|$?

Применение сложности объектов (колмогоровской сложности)

- Случайность
- Бросание монеты
- 011010010101111001001010…
- 010101010101010101010101…
- Вторая последовательность неслучайна?
- Вероятность

одинакова.

- Сложность

разная.

- Последовательность случайна, если ее (колмогоровская) сложность – максимальна.
- Информация в одном объекте о другом…

Сложность вычислений

- О. *Сложность (временная) вычисления* – это число шагов вычисляющего алгоритма.
- Какими могут быть отдельные шаги? Какова «модель вычислений»?
- Например, алгоритмы Маркова
- Задача с разными исходными данными. При конкретном исходном данном можно «запомнить ответ».
- *Реально решаемая задача* – сложность вычисления ограничена полиномом от размера исходного данного (длины двоичного слова). Класс $\mathcal{P}$.
- В реальных задачах коэффициенты и степени полиномов оказываются «небольшими».
- Бывает, что алгоритм очень просто описывается, но требует для своего выполнения много времени. Часто «много» означает экспоненту от размера исходного данного или еще больше.

Задачи, решаемые перебором

- Типичная ситуация (считаем, что объекты – двоичные слова и их размер – это длина):

Задача о рюкзаке. Дано n+1 натуральное число: a_0, a_1,…, a_{n-1}, b, можно ли составить из a_i сумму, равную b?

- Если бы нужно было перебирать только пары a_i, то мы бы нашли нужную комбинацию за полиномиальное время – число пар квадратичное, умножить на время проверки и т. д.
- Берутся не пары, а подмножества – время перебора экспоненциально.

Выполнимость. Дана формула логики высказываний. Выполнима ли она?

- Можно выяснить, перебирая все возможные наборы значений логических имен.
- Разных имен в формуле может быть, по порядку, например, корень квадратный от размера формулы.
- Экспонента может быть не от размера, а от размера, деленного на логарифм размера, или от корня кубического размера, но это не помогает.

Задачи, решаемые перебором

Общая формулировка

• Дано двуместное отношение R(x,y), где x – исходное данное, y – перебираемое (подсказка, подтверждение).

• ***Задача***: выяснить, по данному x, существует ли y, для которого R(x,y):

$$\exists y\ R\ (x,y),$$

причем:

- Размер y ограничен заданным полиномом от размера x.
- Сложность вычисления R(x,y) ограничена полиномом от размера x.
- Отношение R и ограничивающие полиномы фиксированы для данной задачи.

• Задача о рюкзаке и Выполнимость – таковы.

• NP – класс задач, решаемых перебором.

Универсальная переборная задача

- **Универсальный алгоритм** A_0 получает на вход x, описание конкретного алгоритма A, задающего какое-то отношение, второй аргумент отношения – y, и применяет A к x и y. Этот алгоритм задает отношение $U_0(z,y)$, мы его применяем к аргументам $\langle x,p\rangle,y$, где p – описание алгоритма A, получаем

  $$U_0(\langle x,p\rangle,y),$$

  если первый аргумент не оказывается парой такого вида, то отношение U_0 – ложно.
- Умея решать задачу $\exists y U_0(\langle x,p\rangle,y)$, мы сможем решить и любую задачу, решаемую перебором, причем время работы универсального алгоритма будет не сильно отличаться от времени работы алгоритма A.
- Отношение U_0 может и не вычисляться за полиномиальное время: для разных p полиномы, ограничивающие время работы и длину y, могут иметь разную степень.

Универсальная переборная задача

- Модификация: отношение U, задаваемое, как U(< x,p,L >, y), где алгоритм, вычисляющий U, работает так же, как U_0, но при этом его время работы ограничено длиной слова – третьего элемента тройки, являющейся первым аргументом U (если первый аргумент – не такая тройка, то U – ложно).
- U имеет полиномиальную сложность.
- ∃y U(< x,p,L >, y) – *универсальная переборная задача.*

Универсальность

- Предположим, что мы нашли алгоритм, позволяющий решать универсальную задачу ∃y U(z,y) за полиномиальное время.
- Этот алгоритм:
- НЕ перебирает y и
- НЕ применяет U, а делает что-то совсем другое.
- Тогда мы сможем и любую переборную задачу решать за полиномиальное время, сооружая каждый раз тройку $< x,p,L >$, (и при этом не имитируя работу алгоритма с описанием p).

Естественные переборные задачи

- Предположим, что мы можем решать задачу о рюкзаке или “Выполнимость” за полиномиальное время.
- Поможет ли это в решении других переборных задач?
- Да.

Сведение к выполнимости

- Пусть имеется задача З: $\exists y\ R(x,y)$, решаемая перебором, и отношение R задается алгоритмом А.
- По данному x и А строим исходное данное для задачи выполнимости, то есть формулу Ф.
- Часть исходного данного – это x. Строящаяся формула Ф утверждает, что вычисление алгоритма А на аргументе <x,y> дает 1.
- Формула $\exists z\ \Phi(z)$ означает существование y и существование вычисления на <x,y>, дающего 1.

Построение формулы Ф

- Вычисление – последовательность слов – состояний вычисления. Длина этой последовательности и длина каждого члена в ней ограничены полиномом от длины x (фиксированным для задачи 3).
- Удобно выписывать члены последовательности (например) сверху вниз. Все символы слов можно закодировать в двоичном алфавите.
- Для каждого символа введем свое логическое имя. Число этих имен (переменных в формуле) ограничено полиномом (площадь таблицы, занятой вычислением).
- (Но этого мало, надо, чтобы и размер формулы был ограничен.)

Построение формулы Ф

Конъюнкция условий на:

- Первое слово

  – первым аргументом является конкретное слово x: Конъюнкция утверждений $A \equiv 0$ или 1, фиксирующих значения всех букв в x. (A на часть, соответствующую y, никаких ограничений.) Длина – не более квадрата (число членов конъюнкции – длина x, индексы, скобки…)

- Каждые два последовательных слова: P и Q. Дизъюнкция:

  – Фиксированное слово входит в P на данном отрезке и это первая левая часть правила, которая входит в P, и это первое вхождение этого слова в P;

  – Слово Q совпадает со словом P до указанного вхождения и после него, а на месте вхождения стоит правая часть правила.

- Последнее слово – это 1 и нельзя применить ни одно из правил или на предпоследнем шаге применялось заключительное правило.

Проблема перебора

Доказать, что $P \neq NP$

Проблема равенства классов *P* и *NP* является [первой] из семи задач тысячелетия, за решение которой Математический институт Клэя назначил премию в миллион долларов США.
- Википедия

Колмогоров, Марков, Левин, Хачиян, Разборов