

Тема 5.2. Рекурсивные функции. Нормальные алгоритмы Маркова

Аннотация: Рекурсивные функции. Нормальные алгоритмы Маркова

Рекурсивные функции

Рассмотрим очень коротко еще одну модель уточнения понятия алгоритма – рекурсивные функции.

Всякий алгоритм однозначно сопоставляет допустимым начальным данным результат. Это означает, что с каждым алгоритмом однозначно связана функция, которую он вычисляет. Отсюда возникает вопрос: для всякой ли функции можно указать вычисляющую ее машину Тьюринга, а если нет, то для каких функций существует вычисляющий их алгоритм (машина Тьюринга), как описать такие, как говорят, алгоритмически или эффективно вычисляемые функции?

Исследование этих вопросов привело к созданию в 1930-х гг. теории рекурсивных функций. При этом класс вычисляемых функций (названных здесь рекурсивными) получил такое описание, которое весьма напомнило процесс построения аксиоматической теории на базе некоторой системы аксиом. Сначала были выбраны простейшие функции, эффективная вычислимость которых была очевидна (своего рода «аксиомы»). Затем сформулированы некоторые правила (названные операторами), на основе которых можно строить новые функции из уже имеющихся.

В качестве исходных простейших функций выберем следующие:

$S(x) = x + 1$ (функция следования);

$O(x) = 0$ (нуль-функция);

$I_m^n(x_1, x_2, \dots, x_n) = x_m$ (функции-проекторы, $1 \leq m \leq n$, т.е. функции, результатом которой является один, вполне определенный ее аргумент).

Будем говорить, что n -местная функция φ получена из m -местной функции f и n -местных функций g_1, g_2, \dots, g_m с помощью **оператора суперпозиции**, если для всех x_1, x_2, \dots, x_n справедливо равенство:

$$\varphi(x_1, x_2, \dots, x_n) = f(g_1(x_1, x_2, \dots, x_n), \dots, g_m(x_1, x_2, \dots, x_n)).$$



Если функции $f(x_1, x_2, \dots, x_m), g_1(x_1, x_2, \dots, x_n), \dots,$

Теорема

$g_m(x_1, x_2, \dots, x_n)$ правильно вычислимы по Тьюрингу, то правильно вычислима и сложная функция (суперпозиция функций):

$$\varphi(x_1, x_2, \dots, x_n) = f(g_1(x_1, x_2, \dots, x_n), \dots, g_m(x_1, x_2, \dots, x_n)).$$

Примем эту теорему без доказательства.

Говорят, что $(n+1)$ -местная функция φ получена из n -местной функции f и $(n+2)$ -местной функции g с помощью **оператора примитивной рекурсии**, если для любых x_1, x_2, \dots, x_n, y , справедливы равенства:

$$\varphi(x_1, x_2, \dots, x_n, 0) = f(x_1, x_2, \dots, x_n),$$

$$\varphi(x_1, x_2, \dots, x_n, y+1) = g(x_1, x_2, \dots, x_n, y, \varphi(x_1, x_2, \dots, x_n, y)).$$

Пара этих равенств называется **схемой примитивной рекурсии**.

Функция называется **примитивно рекурсивной**, если она может быть получена из простейших функций O, S, I_m^n с помощью конечного числа применений операторов суперпозиции и примитивной рекурсии.

Наконец, введем заключительный, третий, оператор.

Определение 33.5 (оператор минимизации). Будем говорить, что n -местная функция φ получается из $(n+1)$ -местных функций f_1 и f_2 с помощью **оператора минимизации**, или **оператора наименьшего числа**, если для любых x_1, x_2, \dots, x_n, y равенство $\varphi(x_1, x_2, \dots, x_n) = y$ выполнено тогда и только тогда, когда величина $\varphi(x_1, x_2, \dots, x_n)$ равна наименьшему значению аргумента y , при котором выполняется это равенство.

Функция называется **частично рекурсивной**, если она может быть получена из простейших функций O, S, I_m^n с помощью конечного числа применений суперпозиции, примитивной рекурсии и оператора минимизации. Если функция всюду определена и частично рекурсивна, то она называется **общерекурсивной**.

Подобно тезису Тьюринга, в теории рекурсивных функций выдвигается соответствующая гипотеза, носящая название тезиса Чёрча:

Числовая функция тогда и только тогда алгоритмически (или машинно) вычислима, когда она частично рекурсивна.

И эта гипотеза не может быть доказана строго математически, она подтверждается практикой, опытом, ибо призвана увязать практику и теорию. Все рассматривавшиеся в математике конкретные функции, признаваемые вычислимыми в интуитивном смысле, оказывались частично рекурсивными.

Рекурсивные функции также рассматриваются подробно в предмете «Математическая логика и теория алгоритмов».

Нормальные алгоритмы Маркова

Также очень коротко рассмотрим еще одну модель, называемую нормальными алгоритмами Маркова - один из стандартных способов формального определения понятия алгоритма. Понятие нормального алгоритма введено А. А. Марковым (младшим) в конце 1940-х годов в работах по неразрешимости некоторых проблем теории ассоциативных вычислений.

Нормальный алгоритм описывает метод переписывания строк, похожий по способу задания на формальные грамматики. НАМ — полный по Тьюрингу язык, что делает его по выразительной силе эквивалентным машине Тьюринга и, следовательно, современным языкам программирования. На основе НАМ был создан функциональный язык программирования Рефал.

Алфавитом называется любое непустое множество. Его элементы называются **буквами**, а любые последовательности букв — **словами** в данном алфавите. Для удобства рассуждений допускаются пустые слова. Пустое слово будем обозначать Λ . Если A и B — два алфавита, причем $A \subseteq B$, то алфавит B называется **расширением** алфавита A .

Слова будем обозначать латинскими буквами: P, Q, R . Одно слово может быть составной частью другого слова. Тогда первое называется **подсловом** второго или вхождением во второе.

Марковской подстановкой называется операция над словами, задаваемая с помощью упорядоченной пары слов (P, Q) , состоящая в следующем. В заданном слове R находят первое вхождение слова P (если таковое имеется) и, не изменяя остальных частей слова R , заменяют в нем это вхождение словом Q . Полученное слово называется результатом применения марковской подстановки (P, Q) к слову R . Если же первого вхождения P в слово R нет, то считается, что марковская подстановка (P, Q) неприменима к слову R .

Для обозначения марковской подстановки (P, Q) используется запись $P \rightarrow Q$. Она называется формулой подстановки (P, Q) . Для заключительной подстановки будем использовать обозначение $P \rightarrow .Q$

Упорядоченный конечный список формул подстановок

$$\begin{cases} P_1 \rightarrow (.)Q_1 \\ P_2 \rightarrow (.)Q_2 \\ \dots \\ P_n \rightarrow (.)Q_n \end{cases}$$

в алфавите A называется схемой (или записью) нормального алгоритма в A . (Запись точки в скобках означает, что она может стоять в этом месте, а может отсутствовать.) Данная схема определяет (детерминирует) алгоритм преобразования слов, называемый нормальным алгоритмом Маркова. Дадим его точное определение.

Нормальным алгоритмом (Маркова) в алфавите A называется следующее правило построения последовательности V_i слов в алфавите A , исходя из данного слова V в этом алфавите. В качестве начального слова V_0 последовательности берется слово V . Пусть для некоторого $i \geq 0$ слово V_i построено и процесс построения рассматриваемой последовательности еще не завершился. Если при этом в схеме нормального алгоритма нет формул, левые части которых входили бы в V_i , то V_{i+1} полагают равным V_i , и процесс построения последовательности считается завершившимся. Если же в схеме имеются формулы с левыми частями, входящими в V_i , то в качестве V_{i+1} берется результат марковской подстановки правой части первой из таких формул вместо первого вхождения ее левой части в слово V_i ; процесс построения последовательности считается завершившимся, если на данном шаге была применена формула заключительной подстановки) и продолжающимся — в противном случае. Если процесс построения упомянутой последовательности обрывается, то говорят, что рассматриваемый нормальный алгоритм применим к слову V . Последний член W последовательности называется результатом применения нормального алгоритма к слову V .

Пример: Пусть $A = \{a, b\}$ — алфавит. Рассмотрим следующую схему нормального алгоритма в A :

$$\begin{cases} a \rightarrow .\Lambda \\ b \rightarrow b. \end{cases}$$

Тогда получим, например, следующие соотношения:

$$aaba \Rightarrow aba,$$

$$aaaa \Rightarrow aaa,$$

$$bbaaba \Rightarrow bbaba,$$

$$bbba \Rightarrow bbb,$$

$$bababab \Rightarrow bbabab.$$

Т.е. всякое слово в алфавите A , содержащее хотя бы одно вхождение буквы a , он перерабатывает в слово, получающееся вычеркиванием в нем самого левого (первого) вхождения буквы a . Пустое слово он перерабатывает в пустое. (Алгоритм не применим к таким словам, которые содержат только букву b .)

Как и машины Тьюринга, нормальные алгоритмы не производят собственно вычислений: они лишь производят преобразования слов, заменяя в них одни буквы другими по предписанным им правилам. В свою очередь, мы предписываем им такие правила, результаты применения которых мы можем интерпретировать как вычисления.

Функция f , заданная на некотором множестве слов алфавита A , называется **нормально вычислимой**, если найдется такое расширение B данного алфавита ($A \subseteq B$) и такой нормальный алгоритм в B , что каждое слово V (в алфавите A) из области определения функции f этот алгоритм перерабатывает в слово $f(V)$.

Создатель теории нормальных алгоритмов советский математик А. А. Марков выдвинул гипотезу, получившую название «**Принцип нормализации Маркова**». Согласно этому принципу, для нахождения значений функции, заданной в некотором алфавите, тогда и только тогда существует какой-нибудь алгоритм, когда функция нормально вычислима.

Вопросы для самоконтроля:

1. Рекурсивные функции. Тезис Черча.
2. Нормальные алгоритмы Маркова. Примеры.