Курсовая работа

по дисциплине: Математическая логика и теория алгоритмов

на тему: Рекурсивные функции

Содержание

Введение

1. Цели и задачи курсовой работы

2. Алгоритмическая неразрешимость

3. Суперпозиция функций

3. Рекурсивные функции

3.1 Примитивно рекурсивная функция

3.2 Общерекурсивные функции

3.3 Расширенный базис клини. Частичная рекурсивность разности

3.4 Геделева нумерация и эффективная перечислимость частично-рекурсивных функций

Заключение

Список использованной литературы

Введение

Важным элементом математики является доказательство существования или отсутствия алгоритма для решения поставленной задачи.

Такое доказательство позволяет однозначно ответить на вопрос, существует ли алгоритмическое решение, и таким образом уберегает от бесплодных попыток найти его в случае его отсутствия.

Задача является алгоритмически неразрешимой, если не существует конечной последовательности действий, приводящей к правильному результату при любых начальных условиях.

Существует несколько подходов к определению понятия алгоритма. Одним из них связано с уточнением понятия эффективно вычислимой функции. Его вывели математики А. Черч, К. Гендель, С. Клини. В результате был выведен класс частично-рекурсивных функций, имеющих строгое математическое определение.

1. Цели и задачи курсовой работы

В данной курсовой работе рассматриваются: определение алгоритмической неразрешимости задачи; понятия суперпозиции функций, рекурсивных функций; схема примитивной рекурсии; операция минимизации.

Целью курсовой работы является раскрытие понятий алгоритмически вычислимой функции, рекурсивной функции.

Задачами курсовой работы являются:

. пояснение определения алгоритмической проблемы, алгоритмической неразрешимости;

. раскрытие понятия суперпозиции функций;

. выяснение роли частично рекурсивных и общерекурсивных функций в определении вычислимости функций; анализ схемы примитивной рекурсии.

2. Алгоритмическая неразрешимость

алгоритм задача рекурсия суперпозиция

Алгоритмическая неразрешимость - важнейшее свойство некоторых классов корректно поставленных задач, допускающих применение алгоритмов. Оно состоит в том, что задачи каждого из этих классов в принципе не имеют какого-либо общего, универсального алгоритма решения, объединяющего этот класс. Несмотря на полную однотипность условий и требований, здесь, как ни парадоксально, принципиально невозможна однотипность метода решения. А. н. не означает неразрешимости тех или иных единичных проблем данного класса - часть из них может иметь свои решения. Но в целом данный класс задач не имеет ни общего универсального алгоритма решения, ни ветвящегося алгоритма полного разбиения класса на подклассы, к каждому из которых был бы применим свой специфический алгоритм.

Алгоритмически неразрешимыми являются, напр., проблема распознавания: закончит ли свою работу (остановится ли) или же «зависнет» в бесконечном цикле произвольно выбранная программа действий алгоритмического типа (не только компьютерная, но и реализуемая человеком по алгоритмическому типу); проблема эквивалентности программ (нет универсального алгоритма, позволяющего установить эту эквивалентность); проблема тождества двух математических выражений; проблема распознавания того, можно ли из имеющихся автоматов собрать заданный автомат; а также множество других проблем, относящихся к топологии, к теории групп и к другим областям.

3. Суперпозиция функций

Суперпозицией функций f1, …, fm называется функция f, полученная с помощью подстановок этих функций друг в друга и переименования переменных.

Пусть имеются два отображения <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp=f : X\_f \rightarrow Y\_f> и <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp=g : X\_g \rightarrow Y\_g>, причем <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp= f (X\_f) \cap X\_g>непустое множество. Тогда супер позицией или композицией функций <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp=g> и <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp=f>называется функция <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp= g \circ f : X\_f \rightarrow Y\_g>, определенная равенством <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp= g \circ f(x) = g(f(x))> для всякого <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp=x \in X\_f>.

Областью определения суперпозиции является множество <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp=\mathbb{D}\_h = \left \{ x \in X\_f \mid f(x) \in X\_g \right \}>.

Функция <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp=g> называется внешней, а <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp=f>-внутренней функцией для суперпозиции <http://moodle.herzen.spb.ru/filter/tex/displaytex.php?texexp=g\circ f>.

Функции, представленные в виде композиции "более простых", называются сложными функциями.

Примерами использования суперпозиции являются: решение системы уравнений методом подстановки; нахождение производной от функции; нахождение значения алгебраического выражения с помощью подстановки в него значения заданных переменных.

3. Рекурсивные функции

Рекурсией называется такой способ задания функции, при котором значения определяемой функции для произвольных значений аргументов выражаются известным образом через значения определяемой функции для меньших значений аргументов.

.1 Примитивно рекурсивная функция

Определение понятия примитивно рекурсивной функции является индуктивным. Оно состоит из указания класса базовых примитивно рекурсивных функций и двух операторов (суперпозиции и примитивной рекурсии), позволяющих строить новые примитивно рекурсивные функции на основе уже имеющихся.

К числу базовых примитивно рекурсивных функций относятся функции следующих трёх видов:

Нулевая функция- функция без аргументов, всегда возвращающая<https://ru.wikipedia.org/wiki/0\_(%D1%87%D0%B8%D1%81%D0%BB%D0%BE)>.

Функция следованияодного переменного, сопоставляющая любому натуральному числунепосредственно следующее за ним натуральное число .

Функции , где , от n переменных, сопоставляющие любому упорядоченному набору натуральных чисел число из этого набора.

Операторы подстановки и примитивной рекурсии определяются следующим образом:

Оператор суперпозиции (иногда-оператор подстановки). Пусть - функция от m переменных, а - упорядоченный набор функций отпеременных каждая. Тогда результатом суперпозиции функцийв функцию называется функцияотпеременных, сопоставляющая любому упорядоченному набору натуральных чисел число .

Оператор примитивной рекурсии. Пусть - функция от n переменных, а - функция от переменных. Тогда результатом применения оператора примитивной рекурсии к паре функций и называется функция от переменной вида ;

.

В данном определении переменнуюможно понимать как счётчик итераций, - как исходную функцию в начале итерационного процесса, выдающего некую последовательность функцийпеременных, начинающуюся с , и - как оператор, принимающий на входпеременных , номер шага итерации , функцию на данном шаге итерации, и возвращающий функцию на следующем шаге итерации.

Множество примитивно рекурсивных функций - это минимальное множество, содержащее все базовые функции и замкнутое относительно указанных операторов подстановки и примитивной рекурсии.

В терминах императивного программирования - примитивно рекурсивные функции соответствуют программным блокам, в которых используется только арифметические операции, а также условный оператор и оператор арифметического цикла (оператор цикла, в котором число итераций известно на момент начала цикла). Если же программист начинает использовать оператор цикла while, в котором число итераций заранее неизвестно и, в принципе, может быть бесконечным, то он переходит в класс частично рекурсивных функций.

Укажем на ряд широко известных арифметических функций, являющихся примитивно рекурсивными.

Функция сложения двух натуральных чисел () может быть рассмотрена в качестве примитивно рекурсивной функции двух переменных, получаемой в результате применения оператора примитивной рекурсии к функциям и , вторая из которых получается подстановкой основной функции в основную функцию :

;

;

.

Умножение двух натуральных чисел может быть рассмотрено в качестве примитивно рекурсивной функции двух переменных, получаемой в результате применения оператора примитивной рекурсии к функциям и , вторая из которых получается подстановкой основных функций и в функцию сложения:

;

;

.

Симметрическая разность(абсолютная величина разности) двух натуральных чисел () может быть рассмотрена в качестве примитивно рекурсивной функции двух переменных, получаемой в результате применения следующих подстановок и примитивных рекурсий:

;

;

;

;

;

.2 Общерекурсивные функции

Функция является общерекурсивной, если она определена посредством ряда уравнений некоторого типа.

Затруднение вызвано тем, что просмотр такого выбора уравнений обычно не убеждает нас в том, что эти уравнения “определяют” какую-либо функцию вообще. Когда говорится “функция”, обычно понимается под этим “правило, которое каждому значению (или набору из п значений, если мы имеем функцию n переменных) ставит в соответствие результирующее значение”. Но в общем случае система уравнений не всегда дает определенное значение.

Т.о. вполне допустимой является ситуация, когда для некоторых точек (или даже для всех) значение функции не определено - считается, что в этих точках сама функция не определена. Это явление весьма характерно. Поэтому, делая предположение, что система уравнений действительно определяет общерекурсивную функцию, нужно всегда проявлять осторожность. Мы обычно требуем дополнительного доказательства этого положения, например, в виде индуктивного доказательства того, что для каждого значения аргумента вычисление завершается выдачей единственного значения.

Вследствие этого будем обычно говорить о частично-рекурсивной функции. Когда мы говорим о частично-рекурсивной функции (или, короче, частичной функции) F(x), то понимаем, что может не существовать значения, определенного для некоторого (или даже любого!) значения х. Если F(x) оказывается определенной для всех значений х, то мы называем ее общерекурсивной функцией или, короче, общей функцией. Конечно, любая общерекурсивная функция также может считаться частично-рекурсивной.

Итак, в общем случае, произвольная система рекурсивных уравнений задает именно частично - рекурсивную функцию.

Тезис Черча: класс алгоритмически (или машинно) вычислимых частичных числовых функций совпадает с классом всех частично рекурсивных функций.

Иными словами, арифметическая функция, вычислимая в общем смысле, есть суть рекурсивная функция. Соотв. частичная арифметическая функция, вычислимая в общем смысле, есть суть частично рекурсивная функция. Это не строгое определение, это вывод на уровне тезиса (не доказанное, но и не опровергнутое утверждение).

.3 Расширенный базис Клини. Частичная рекурсивность разности

Частичная арифметическая функция f называется частично рекурсивной, если она может быть получена из простейших функций O, S, Umn конечным числом операций подстановки, примитивной рекурсии и минимизации.

Итак, для того, чтобы задать частично-рекурсивную функцию в виде элементарной схемы ее вычисления, необходимо расширить введенный ранее базис Клини за счет появления новой операции: операции минимизации.

Введем понятие оператора минимизации.

Пусть имеется функция f(x1,….xn), принадлежащая к множеству частичных арифметических функций (ЧАФ). Пусть существует какой то механизм ее вычисления, причем значение функции неопределенно тогда и только тогда, когда этот механизм работает бесконечно, не выдавая никакого определенного результата.

Фиксируем какие-нибудь значения x1,......,xn-1 для первых n-1 аргументов функции f и рассмотрим уравнение: f(x1,.....,xn-1,y)=xn. (1).

Чтобы найти решение y (натуральное) этого уравнения, будем вычислять при помощи указанного выше "механизма" последовательно значения f(x1,...,xn-1,a)=xn, для y=0,1,2,... Наименьшее значение a, для которого получится f(x1,...xn-1,a)=xn, обозначим через мy(f(x1,...,xn-1,y)=xn). (2)

Описанный процесс нахождения выражения (2) будет продолжаться бесконечно в следующих случаях:

Значение f(x1,.....,xn-1,y) не определено;

Значения f(x1,.....,xn-1,y) для y=0,1,...,a-1 определены, но отличны от xn, а значение f(x1,.....,xn-1,a) - не определено

Значения f(x1,.....,xn-1,y) определены для всех y=0,1,2,... и отличны от xn.

Во всех этих случаях значение выражения (2) считается неопределенным. В остальных случаях описанный процесс обрывается и дает наименьшее решение y=a уравнения (1).Это решение, как сказано, и будет значением выражения (2).

Например, для функции разности f(x,y)=x-y в согласии с указанным смыслом символа м, для всех x,y имеем: y=мz(y+z=x) (3)

Напротив, значение выражения мy(y(y-(x+1))=0) (4)

неопределенное, так как уже значение терма 0я(0-(x+1)) неопределенное. В то же время уравнение y(y-(x+1))=0 имеет решение y=x+1, но оно не совпадает с значением выражения (4). Этот пример показывает, что для частичных функций f(x1,.....,xn-1,y) выражение (2), строго говоря, не есть наименьшее решение уравнения (1). Если же функция f(x1,.....,xn-1,y) всюду определенная и уравнение (1) имеет решения, то (2) есть наименьшее решение для (1).

Отсюда возникает закономерное желание использовать под оператором минимизации только всюду определенные функции. Наилучший способ убедиться в том что функция всюду определена - использовать заведомо примитивно рекурсивные функции (или функции, примитивная рекурсивность которых уже доказана ранее или вытекает из способа их задания). В этом случае значения f(x1,.....,xn) будет неопределенно только в одном из трех способов, а именно если значения f(x1,.....,xn-1,y) определены для всех y=0,1,2,... и отличны от xn. Для этого имеющуюся функцию нужно преобразовать таким образом (путем переноса слагаемых, умножения обеих частей, и т.д.) чтобы по обе стороны равенства стояли примитивно рекурсивные (а значит всюду определенные) выражения. В общем виде, под оператором минимизации получим некое рекурсивное уравнение, части которого зависят от переменных (x1,....., xn-1, xn, y), которое принимает значение «истина» или «ложь» при последовательно выбираемых значениях параметра y.

Частичная функция f называется частично рекурсивной, если она может быть получена из простейших функций O,S, Umn конечным числом операций подстановки, примитивной рекурсии и минимизации.

Из указанного основного определения непосредственно вытекают следующие свойства относительно рекурсивных функций.

Каждая частичная функция, примитивно рекурсивная относительно системы функций у, является и частично рекурсивной относитеьно у. В частности, все примитивно рекурсивные функции частично рекурсивны.

Класс частично рекурсивных функций шире класса примитивно рекурсивных функций, так как все примитивно рекурсивные функции всюду определенные, а среди частично рекурсивных функций встречаются и функции не всюду определенные, например нигде не определенная функция (x)=мz(x+1+z=0).

Операции подстановки, примитивной рекурсии и минимизации, произведенные над функциями, частично рекурсивными относительно системы у, дают в результате функции, снова частично рекурсивные относительно у.

3.4 Геделева нумерация и эффективная перечислимость частично-рекурсивных функций

По некоторым причинам нам хотелось бы иметь упорядоченную последовательность (список) всех частично-рекурсивных функций 1 (x), f2 (x), …fn (x), …,

тaк, чтобы о ней можно было говорить на языке эффективно определенных операций. Упорядочение множества объектов в виде определенной дискретной последовательности называется перечислением множества объектов. Для наших целей важно, чтобы перечисление частично-рекурсивных функций было эффективным, т.е. чтобы у нас был эффективный способ нахождения n-й функции списка или хотя бы нахождения ее определения. Хотя это может показаться странным, но нам будет достаточно знать, что перечисление существует - нам даже не потребуется выяснять какие-либо подробности о его свойствах! Следовательно, вместо подробного доказательства достаточно привести убедительный довод в пользу его возможности, т.е. что мы могли бы построить его, если бы это потребовалось.

Почему мы все же могли предположить, что существует эффективное перечисление частично-рекурсивных функций? В конце концов, они образуют фантастически богатое, причудливое и неупорядоченное множество объектов, поскольку в это множество входят объекты, соответствующие всем возможным рекурсивным определениям.

На самом деле при ближайшем рассмотрении ясно, что определение каждой частично-рекурсивной функции состоит из конечного набора уравнений, каждое из которых включает конечное число символов, обозначающих нуль, тождество, следование, примитивную рекурсию, суперпозицию и наименьшее число в сочетании с конечным числом запятых и скобок. Точные правила их расположения не важны; важно то, что в любой такой ситуации всегда можно ввести некоторого рода бесконечное лексикографическое упорядочение составных объектов. Хотя эта ситуация кажется более сложной, есть способ сделать ее даже проще. Действительно, поскольку большинство частично-рекурсивных функций (или, скорее, определений) вовсе не определяет истинного окончания процесса вычислений, нам не надо слишком заботиться, чтобы при перечислении определений все последовательности символов были правильными. Действительно, мы можем рассмотреть какую-либо процедуру, которая в конце концов образует любую (и все) последовательности символов, требуемых для определений. Имеется только одна техническая проблема - остаться в рамках фиксированного алфавита; и она решается использованием, скажем, двоичного кодирования неограниченного числа названий переменных и функций, которые могут потребоваться. Тогда простая схема числового упорядочения будет эффективным перечислением при условии, что существует способ правильной интерпретации n-го описания.

Бесконечное лексикографическое упорядочение составных объектов легко нумеруется при помощи алгоритма Геделя. Кроме того, из теоремы Клини следует, что частично-рекурсивные функции могут быть представлены в виде операторного терма, т.е. в виде схемы частичной рекурсии, в которой используется всего несколько функциональных символов: для трех функций O, S, Umn и трех операций: S с индексами для подстановки, R для примитивной рекурсии и м для минимизации. Учитывая этот факт снимается вопрос о правильной интерпретации n-го описания: если кодировать будем не системы уравнений, а схемы частично-примитивных рекурсий в виде операторных термов, то такая интерпретация будет автоматической.

Построим доказательство эффективной перечислимости на основе геделевой нумерации. Для этого надо ввести кодовые номера для всех элементов при помощи геделевой нумерации.

Возьмем ряд простых чисел: 2, 3, 5, 7,… Дальнейшая процедура аналогична геделевой нумерации, например, алгоритмов Маркова. Т.о. мы получили геделев номер примитивно-рекурсивного описания. Геделевы номера эффективно распознаются среди натуральных чисел описания можно эффективно перечислить.

Некоторые функции fn {x) не могут быть определены даже хотя бы для одного значения x. Некоторые функции с разными индексами, например fi (x) и fj (x), могут быть одинаковыми: fi (x) = fj (x) для всех x. Действительно, некоторые функции могут появляться в списке бесконечно часто, что соответствует совершенно разным определениям. Единственное, в чем мы уверены, это то, что если функция частично-рекурсивна, то она имеет по крайней мере один индекс в перечислении.

Частичная рекурсивность нигде не определенной функции и функции, определенной в одной точке.

1) Нигде не определенная функция .

Построим рекурсивное уравнение: 

Результат операции минимизации не определен даже для точки х=0.

2) Функция, определенная в одной точке, .

Построим рекурсивное уравнение: 

Результат операции минимизации определен только для точки х=0, при остальных значениях x вычисление (подбор нужного значения z) никогда не будет закончено.

Заключение

В данной курсовой работе рассматриваются: определение алгоритмической неразрешимости задачи; понятия суперпозиции функций, рекурсивных функций; схема примитивной рекурсии; операция минимизации.

Целью курсовой работы является раскрытия понятий алгоритмически вычислимой функции, рекурсивной функции.

Задачами курсовой работы являются: пояснение определения алгоритмической проблемы, алгоритмической неразрешимости; раскрытие понятий суперпозиции функций; выяснение роли частично рекурсивных и общерекурсивных функций в определении вычислимости функций; анализ схемы примитивной рекурсии.

В ходе выполнения курсовой работы были изучены и закреплены знания по темам: понятий алгоритмически вычислимая функция, рекурсивная функция.

Были рассмотрены основные понятия заданной темы:

алгоритмическая проблема

алгоритмическая неразрешимость

суперпозиция функций

примитивная рекурсия

частично рекурсивные и общерекурсивные функции.

В представленной работе был проведен анализ понятий: алгоритм решения задачи, алгоритмическая неразрешимость, суперпозиция функций, рекурсивная функция.

Все задачи, поставленные в задании к данной курсовой работе, выполнены в полном объеме.

Вывод:

Доказательство существования или отсутствия алгоритма для решения поставленной задачи является важным элементом математической логики, так как оно позволяет однозначно ответить на вопрос, существует ли алгоритмическое решение.

Задача является алгоритмически неразрешимой, если не существует конечной последовательности действий, приводящей к правильному результату при любых начальных условиях.

Одним из способов такого доказательство является определения принадлежности функции к классу частично-рекурсивных функций.

Список использованной литературы

1. Клини С. Математическая логика. - М.: Мир, 1980.

2. Карри Хаскелл Б. Основания математической логики. - М.: Мир, 1969

. Карпов В.Г., Мощенский В.А. Математическая логика и дискретная математика. - Минск: «Вышэйш. школа», 1977

. Верещагин Н.К., Успенский В.А., Плиско В.Е. Вводный курс математической логики: учебное пособие - ФИЗМАТЛИТ, 2007

. Ершов Ю.Л., Палютин Е.А. Математическая логика - ФИЗМАТЛИТ, 2011

. Зайцева Е.В., Гурова Л.М. Математическая логика и теория алгоритмов: Учебное пособие - Издательство Московского государственного