Отчет по программе

Концевая Маргарита, ИУ8-54, 30 вариант задания

Оглавление

Задание

Реализуйте программу аналитического вычисления производной функции

Теоретическая часть

Вычисление производной производится по следующим правилам:

1) Производная суммы равна сумме производных

$$(u \pm v)' = u' \pm v'$$

2) Производная произведения функций

$$(uv)' = u'v + uv'$$

3) Производная частного функций

$$\left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$$

4) Производная сложной функции

$$(u(v))' = u'(v) \cdot v'$$

ПРОИЗВОДНЫЕ ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ

Функция	Производная
$f(x) = c$	$c' = 0$, где c — const
$f(x) = x^\alpha$	$(x^\alpha)' = \alpha x^{\alpha-1}$
$f(x) = e^x$	$(e^x)' = e^x$
$f(x) = a^x$	$(a^x)' = a^x \ln a$
$f(x) = \ln x$	$(\ln x)' = \frac{1}{x}$
$f(x) = \log_a x$	$(\log_a x)' = \frac{1}{x \ln a}$
$f(x) = \sin x$	$(\sin x)' = \cos x$
$f(x) = \cos x$	$(\cos x)' = -\sin x$
$f(x) = \operatorname{tg} x$	$(\operatorname{tg} x)' = \frac{1}{\cos^2 x}$
$f(x) = \operatorname{ctg} x$	$(\operatorname{ctg} x)' = -\frac{1}{\sin^2 x}$
$f(x) = \arcsin x$	$(\arcsin x)' = \frac{1}{\sqrt{1-x^2}}$
$f(x) = \arccos x$	$(\arccos x)' = -\frac{1}{\sqrt{1-x^2}}$
$f(x) = \operatorname{arctg} x$	$(\operatorname{arctg} x)' = \frac{1}{1+x^2}$
$f(x) = \operatorname{arcctg} x$	$(\operatorname{arcctg} x)' = -\frac{1}{1+x^2}$

Методы решения

Для решения данной задачи используется автоматическое дифференцирование, которое позволяет вычислить производную на этапе компиляции, но данный метод нам не подходит, т.к. выражение вводится пользователем только на этапе выполнения.

Другим методом нахождения аналитической производной является построение синтаксического дерева решения. В данной работе используется рекурсивный вызов функции производной от строки, которая разбивается на элементарные выражения, которые далее обрабатываются согласно правилам дифференцирования и таблице производных. Данный метод похож на построение синтаксического дерева, которое в данном случае осуществляется выделением элементарных частей и взятие производных от них.

Инструкция по использованию программы

Для использования программы необходимо собрать проект, следуя инструкциям, приведенным на официальном сайте Microsoft (для конкретно вашего оборудования и окружения).

Например:

- ***cl /EHsc deritivate.cpp Differential.cpp***
- ***deritivate.exe output.txt input.txt***

Описание логики работы программы

Программа работает следующим образом:

1. При запуске программы выводится поле для введения входных данных. Данные проверяются на корректность и, если они проходят проверку, то далее вычисляется решение.
2. Решение задачи осуществляется следующим образом:
    - Считанная корректная строка подаётся в рекурсивную функцию ***dif*.**
    - В строке выполняется поиск операторов в следующем порядке: сложение, вычитание, умножение, деление, степень, тригонометрические функции, логарифм.
    - Позиции найденных операторов записываются в отдельные вектора.
    - Выполняется проверка на пустоту векторов позиций в порядке приоритета операторов. В случае непустоты данного вектора в соответствии с правилами дифференцирования вычисляется производная от подстрок, полученных в результате разбиения исходной строки по позициям в порядке приоритетности;
    - Для вычисления производной по правилам, вызывается одна или несколько соответствующих функций:
        - ***mult*** – возвращает производную от произведения
        - ***div*** – возвращает производную от частного
        - ***pow*** – возвращает производную от степенной функции
        - ***logariphm, flog*** – возвращает производную логарифма (вторая функция проверяет, является ли логарифм натуральным и, в зависимости от проверки, вызывает первую функцию, либо возвращает производную от натурального логарифма)
        - ***fcos*** – возвращает производную от косинуса
        - ***fsin*** - возвращает производную от синуса
        - ***ftg*** - возвращает производную от тангенса
        - ***fctg*** - возвращает производную от котангенса
        - ***farcccos*** - возвращает производную от арккосинуса
        - ***farcsin*** - возвращает производную от арксинуса
        - ***farctg*** - возвращает производную от арктангенса
        - ***farcctg*** - возвращает производную от арккотангенса

      Для операции сложения и вычитания отдельных функций не предусмотрено, ввиду простоты выполнения «на месте».
    - Вышеперечисленные действия повторяются до тех пор, пока подстроки не будут представлять собой элементарные функции: в функцию ***dif*** подаётся подстрока исходной строки, вектора позиций перезаполняются, и т.д.
    - Перед выводом в файл полученная строка подаётся в функцию ***handling***, которая выполняет обработку вывода:
        - убирает двойные знаки: --, +-, -+
        - убирает умножение на «1»
        - убирает плюс, если он стоит первым символом

3. В конце работы алгоритма решения получаем выходные данные: строка с вычисленной производной функции.

Используемые структуры данных

В программе в качестве структуры данных были использованы векторы, т.к.

- Простой доступ к любому элементу (по индексу)
- Нет необходимости удаления и сортировки элементов

Таким образом, вектор – оптимальная структура данных.

Формат входных и выходных данных

Входные данные

В командную строку при запуске программы подаются названия двух файлов: выходного и входного.

Входной файл содержит строки, содержащие функции от x, от которых нужно взять производную.

Используемые знаки операций: +, -, *, /, ^, (,).

Элементарные функции записываются в виде: cos(x), sin(x), tg(x), ctg(x), log(a, x), ln(x) - где x может быть также функцией, зависящей от x, a – основание логарифма.
Константы: e – экспонента, pi – число Пи.

Выходные данные

В выходной файл записываются строки, содержащие вычисленные производные от исходных функций.

Тесты

К программе приложены тесты:

Тесты, проверяющие корректность работы программы на элементарных функциях:

1. Plus.in - примеры сложения элементарных функций
2. Minus.in –примеры вычитания элементарных функций
3. Mult.in – примеры произведения элементарных функций
4. Div.in – примеры частного элементарных функций
5. Pow.in – примеры возведения в степень элементарных функций
6. Log.in – примеры логарифмов от элементарных функций
7. Cos.in – примеры косинусов
8. Sin.in – примеры синусов
9. Tg.in – примеры тангенсов
10. Ctg.in – примеры котангенсов
11. Arcsin.in –примеры арксинусов
12. Arccos.in – примеры арккосинусов
13. Arctg.in – примеры арктангенсов
14. Arcctg.in – примеры арккотангенсов

Тест, проверяющий работу программы на сложных функциях

Complicated.txt – примеры композиций элементарных функций (функция от функции)

Тест, проверяющий корректную обработку некорректного ввода:

Incorrect.txt – незакрытые скобки, опечатки, отсутствующие символы, лишние символы

В результате проверки тестами установлено, что программа работает корректно. Для запуска тестов файл *test.cpp*

Оценка программы

Оценка сложности

Сложность зависит от количества вызовов функции взятия производной dif, которое в свою очередь зависит от количества операторов во вводимой строке. Обозначим количество операторов числом n, длину строки – числом l. Каждый оператор вызывает функцию dif дважды: для взятия производной от него самого и от его аргументов. Таким образом, функция вызывается 2n раз в худшем случае. Сложность самой функции –O(l), т.к. мы проверяем строку посимвольно.

Сложность в целом тогда: O(l*n).

Оценка используемой памяти

Программа хранит в себе строку, исходная длина которая = l, и на каждом шаге функции хранит новую строку, которая является частью исходной. Строка будет разбиваться на подстроки n раз, тогда сложность по памяти составит: O(l*n).