$CalCul^2$

-- Calculus Calculating Language

Junde Huang (jh3419)
Kewei Ge (kg2481)
Wenting Yin (wy2214)
Zhan Shu (zs2242)
Jinxi Zhao (jz2540)

1 Introduction

1.1 Motivation

$CalCul^2$ is a programming language focusing on math calculation such as calculus. Like a powerful calculator, $CalCul^2$ can calculate values, deal with math functions, and even do calculus. It is very useful and convenient.
Although the existing languages, such as C or Java, could also handle math computations, they often requires complicated programming, which is very tedious for users. Our $CalCul^2$ will allow users to describe and manipulate calculus in a much simpler way.

1.2 Language Description

$CalCul^2$ is a simple yet powerful interpreted language that deals with limits and the differentiation and integration of functions of one or more variables. $CalCul^2$ supports basic mathematical as well as calculus operations, which involves ordinary differential equation and definite integral calculation.Instead of redundancy types, developers can solve problems with clear and simple codes in $CalCul^2$. E.g. To make codes concise, $CalCul^2$ does not require a type definition for each variable, even a function, instead it will automatically recognize the type and might adjust it with the processing of programs. Furthermore, $CalCul^2$ also provides ability to simple evaluation of functions. $CalCul^2$ is a concise language for calculus: just using less to accomplish more.

2 Language Definition

2.1 Data Types

Calculus Calculator Language don't need the user to designate data type for every variable. It will be determined by the initial definition. The build-in data types are as follow:

Type	Definition

int	integer value
float	floating-point value
bool	boolean value
vector	a sequence of int, float,bool or vector
function	a mathematical function with several variables

2.2 Basic Arithmetic Operators

As a calculating language we definitely need to bring users the basic arithmetic operators, including '+', '-', '*', '/', '^'(power), '//'(integer division), '%'(module). And also we provide logical operators like '<', '>', '<=', '>=', '==', 'and', 'or' and 'not'.

2.3 Advanced Arithmetic Operators

What we bring to you is not only a simple calculator, but also a powerful mathematical program. We will provide advanced arithmetic operators as follow:

Expression	Definition	Return type
f(expr1,expr2,...)	Value of function f at (expr1,expr2,...)	value(float or int)
f'(x)	Derivative of function f with respect to x	function
f'(expr1,expr2,...)	Gradient of function f at (expr1,expr2,...)	vector
y@x(a,b)	Definite integral of y with x in (a,b)	value(float or int)

All these Arithmetic Operators except f(expr1, expr2,...) are only allowed to be applied on pure mathematical functions(i.e. function not include recursions, loops or condition controls).

2.4 Other Arithmetic Functions and Values

We also provide some arithmetic values like 'Pi' and 'e'. The arithmetic functions include 'sqrt','sin', 'cos', 'tan', 'asin', 'acos', 'atan', 'log' and 'ln'.

2.5 Control flow

In $CalCul^2$, we also provide if-else condition controls and while and for loops. They are all very easy to use.

if-else:

```
if (bool_expr) {
   Statement 1
}
else{
   Statement 2
}

loops:
while (bool_expr) {
   Statement;
}

for (val = a to b) {
   statement;
}
```

2.6 Comment

Comments can be used by '##' or '#*-*#'.

```
## single line comment
#* multi-line comment
  second line *#
```

3 Example

```
## Define a variable
x = 3;

## Define a function f(x) and function variable x
f(x) = 3*x;

## Evaluate f(x) = 3*x + 2 such that x = 4, output: f(x)=14
f(x) = 3*x + 2;
:f(4);

## Get derivative of function f(x)=3*x^2 , using " ' "
f(x) = 3*x^2;
:f'(x);            ## Output will be f'(x) = 6*x

## Get the derivative of function f(x1, x2)=2*x1+3*x2 with respect of x1
f(x1,x2) = 2*x1+3*x2;
:f'(x1);           ## Output will be f'(x1) = 2
```

```
## Get the gradient of function f(x1, x2)=2*x1+3*x2 when x1=2 and x2=3
f(x1,x2) = 2*x1+3*x2;
:f'(2,3);           ## Output will be f'(2,3) = [2,3];

## Compute definite integral of f(x) within interval of 3 and 10
f(x) = 3*x^2  + x;
:f@x(3,10);     ## Output will be f@x(3,10)=1018.5

## using recursion and if-else control
f(x){
    if(x==0)
        f=0;
    else
        f=f(x-1)+x;
}
:f(3);    ## Output will be f(3)=6

## using loop control
f(x){
    f = 0;
    while(not x==0){
        f = f + x;
        x = x - 1;
    }
}
:f(3);    ## Output will be f(3)=6
:f'(3);    ## Error, the function f(x) is not a mathematical function
```