# 一元函数表达式微分计算器

## 开发技术简介

### Python语言

使用Python 3.6.1版本

## 系统需求分析与设计

### 系统需求分析

#### 可行性分析

##### 技术可行性

将一元函数表达式看作一种语言,利用形式语言学,微积分等相关知识,在理论上自动求微分可以利用计算机实现

##### 操作可行性

该软件的输入格式即为普通的函数表达式,一键求解,易用性强

##### 技术约束

该软件使用Python语言进行开发,版本为3.6.1

##### 未来可能的需求

多元函数微分计算,一元函数积分计算,多元函数积分计算,

#### 功能需求分析

##### 对常数求导

对常数求导结果应为0

##### 对变量求导

对某变量求对该变量自身的导数为1,对其它变量求导为0

##### 对加法,减法式求导

##### 对乘法求导

##### 对除法求导

### 软件结构设计

#### 分析

该计算器由交互界面与后台计算组成

交互界面包括输入, 修改输入, 显示结果, 显示提示

后台计算包括对表达式求导,对表达式化简

##### 交互界面

[]

##### 后台计算模块

###### 2.1 求导计算

[[求导规则]](https://en.wikipedia.org/wiki/Derivative)

求导算法

表达式由两个操作数以及一个操作符构成,两个操作数又是表达式,最简单的表达式为一个常数或者一个变量.



因此对表达式的求导可以先求解出两个操作数的导数,再根据操作符对应的求导规则,得出结果,而对两个操作数求导,又可以递归为对其自身的两个操做数求导

如上式, 若operator为 “+”, 则E` = E1`+E2`

最终递归到对简单表达式即常数或一个变量求导,对常数求导即为0,对变量自身求导为其系数,对变量求对其它变量的导为0



表达式构造

该求导方法采用了树形表达式结构, 而为了用户输入的简便性, 其输入为线性结构. 而再形式语言学中,语法分析时,为线性的句型构造了树形的分析树, 因此, 为了将线性的输入转化为树形结构, 引入一种表达式文法, 在进行语法分析时, 将线性结构的输入转化为树形结构的表达式, 并同时还可检查输入的合法性.因此表达式构造模块需要用大词法分析器与语法分析器

表达式文法:

[表达式文法]

###### 2.2 表达式化简

表达式化简亦可在运用语法分析完成, 进行归约或推导(自顶向下分析时为推到,自底向上分析时为归约)应用产生式时, 为产生式定义子程序, 对这次归约的两个子节点进行化简. 即相当于将两个子节点各自的子节点看作属性, 进行语义计算.

#### 设计

经由上述分析可得软件结构图:

v

##### \产生式模块

由表达式文法产生式的一组子程序组成,每一个产生式声明一个接口,由依赖该模块的程序提供实现

##### \词法分析器

构造一个自动机实现分词功能,提供分词功能,以识别常数,变量,运算符,函数符号

该模块对输入进行分词,为语法分析器提供输入

##### \语法分析器

采用自底向上的语法分析方法,对表达式进行语法分析,检查表达式合法性

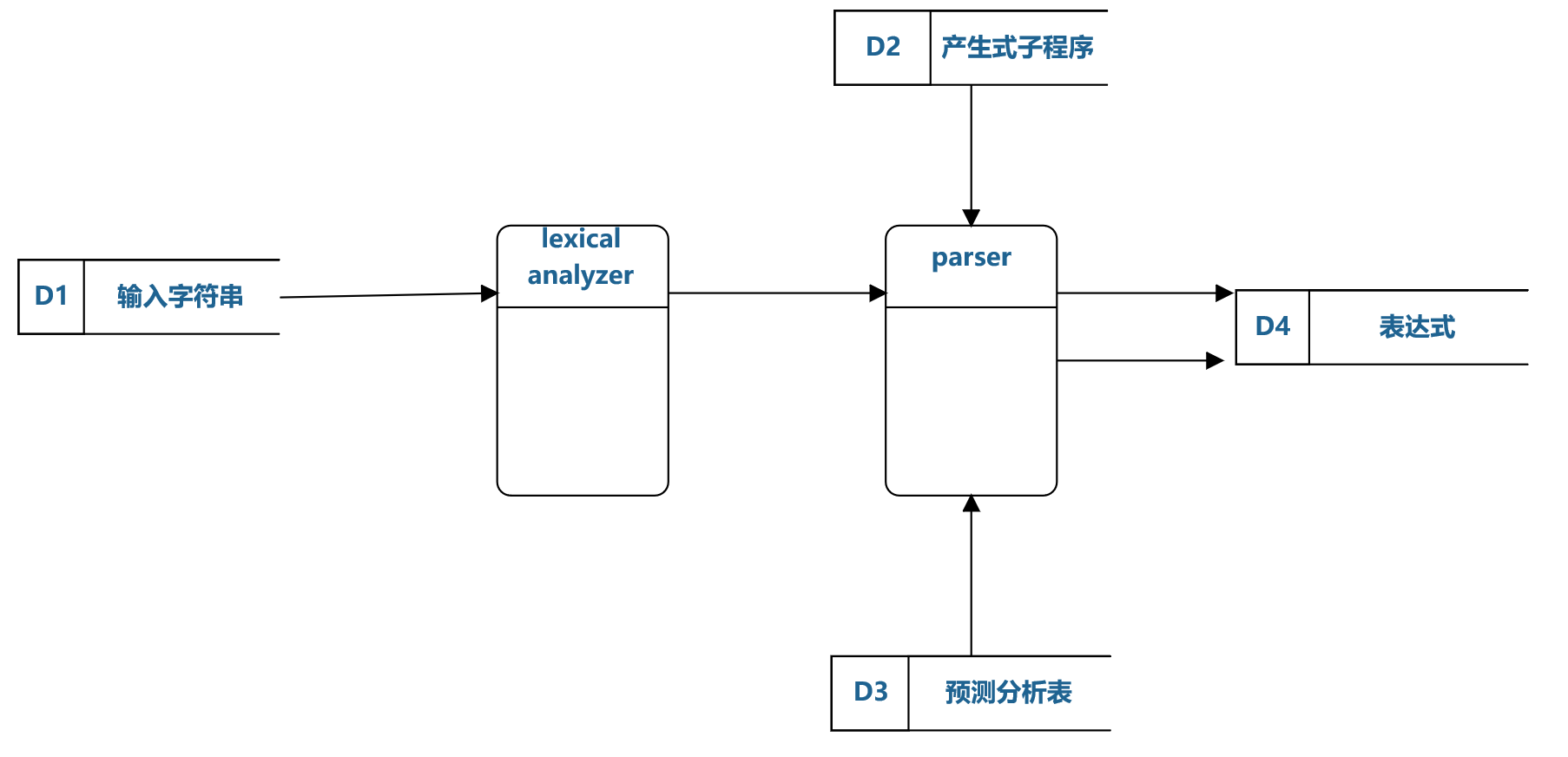
将词法分析器的输出作为输入,并利用产生式模块提供的接口处理输入

##### \表达式构造模块

用户以字符串形式输入表达式,软件接收输入并将其转化为软件内部使用的表达式对象

该模块实现表达式产生模块的相应接口

由产生式模块,词法分析器,语法分析器完成表达式构造



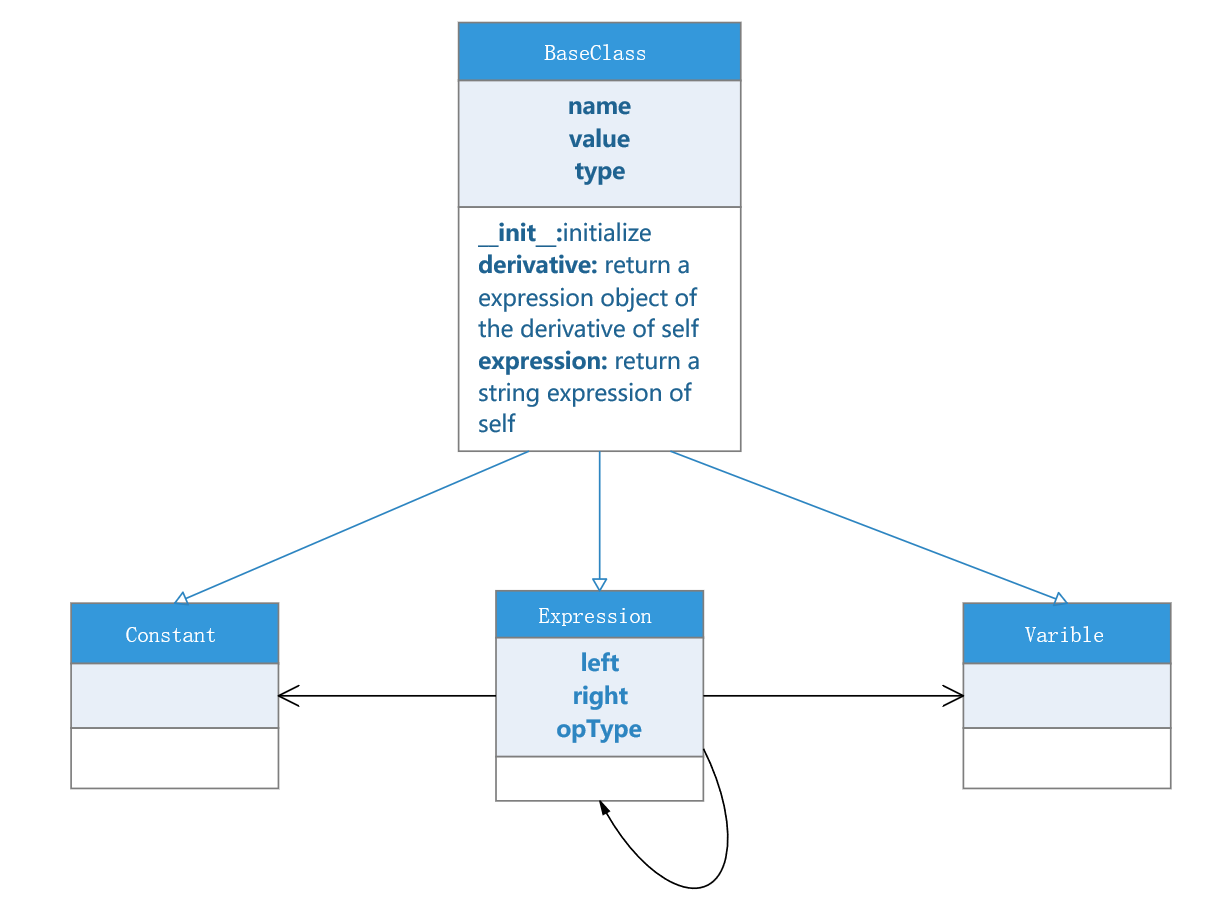
##### 表达式模块

最简单的表达式由常数,变量组成,复杂表达式由表达式通过运算符组成

定义三个对象:常数对象,变量对象,表达式对象

每个对象均实现一个自动求导的接口

因此该软件通过表达式构造模块调用产生式模块,词法分析器,语法分析器构造出表达式,进而自动求导



##### \化简模块

将输入及输出的表达式进行合并同类项等化简

### 软件界面设计

### 模块实现

#### \产生式模块

#### \词法分析器

#### \语法分析器

#### \表达式构造模块

#### 表达式模块

#### \化简模块

### 总结

### 参考文献

### 致谢