

# 扩展通用算术包

信息科学技术学院 王迪 (学号: 1300012802)

2014 年 5 月 4 日

## 1 项目内容

按照 Exercise 完成了如下扩充:

1. **5.1A 5.1B** 定义了通用的`equ?`谓词, 为 `Number` 包添加了`=number`谓词并安装到算术包中。
2. **5.3A 5.3B** 为 `Rational` 包添加了`equ-rat?`谓词并安装到算术包中。
3. **5.4A 5.4B** 定义`repnum->reprat`过程, 并以此实现 `Number` 与 `Rational` 间的四则运算和相等性判断。
4. **5.5B** 定义`create-numercial-polynomial`过程, 通过一个变量符号和一个系数列表生成一个多项式。
5. **5.5C** 补充了`map-terms`的定义, 并定义了通用的`pretty-disp`方法, 以更加友好的方式输出 `Number`、`Rational` 和 `Polynomial`。
6. **5.7A 5.7B 5.7C** 定义了`negate-polynomial`过程, 并以此实现了多项式的减法和相等性判断, 且安装到算术包中。
7. **5.8A 5.8B** 定义`repnum->reppoly`过程, 并以此实现 `Number` 和 `Polynomial` 间的四则运算和相等性判断。
8. **5.9A 5.9B** 补充了`apply-terms`的定义, 使得通用多项式可以进行代入求值。

## 2 基本设计

顺应了提供代码的思路, 可以看出核心思想就是数据导向的程序设计。

通过良好设计的抽象屏障, 不用关心通用算术包的底层数据结构, 通过查看提供的接口就可以方便地进行功能扩充。另外, 使用表格对通用数字的各种运算进行管理, 结构简单、清晰。

## 3 所遇问题

因为提供的代码结构良好, 所以在扩充功能时没有碰到什么大问题。

有一个小问题: 为了进行 `Number` 和 `Rational` 之间的运算, 定义了`NNmethod->NRmethod`过程将一个  $(\text{RepNum}, \text{RepNum}) \rightarrow T$  的过程变为  $(\text{RepNum}, \text{RepRat}) \rightarrow T$  的过程。但这样只能处理`(number rational)`的参数类型, 无法直接处理`(rational number)`的参数类型。

解决方法是写了一个类似的`NNmethod->RNmethod`过程, 这么做是最简单的解决方法, 但一方面这出现了重复代码, 要修改比较繁琐, 另一方面也不易扩展到更多参数时的类型转换。

## 4 改进方法

我认为这个系统最大的缺陷在于进行类型转换很繁琐。一个改进方法就是课本上所说的,为每个类型添加从它到其他类型的转换过程,然后修改`apply-generic`的定义,较为智能地去完成类型转换。