肖飞扬 12335220 统计2班

熊天昊 12335231 统计 6班

吴伟繁 12335210 统计 6班

此次实验我们参考了师兄的的五阶段的程序并对其进行了修改，

首先定义了一个“two”的primitive\_function

define\_primitive\_function "two", two

;; << -- 2 >>

xor rax, rax

inc rax

inc rax

push\_argument\_stack rax

next

目的是在栈的末尾添加2。

然后可以定义一个构造连乘的数列的函数Mul\_series

define\_function "Mul\_series",Mul\_series

xx one,two

xx xswapxx

xx help\_Mul\_series

xx drop

xx end

首先将1,2，3等等入栈，使用one，two函数构成数列首、次项，再使用x|swap|xx函数把输入的最后一个数调整到栈顶，以便进行乘法运算。

为了避免函数过于臃杂，我们还需要定义一个辅助函数help\_Mul\_series帮助实现运算。

define\_function "help\_Mul\_series", help\_Mul\_series

xx dup, one?, false?branch,2

xx end

xx sub1

xx xxoverx

xx multiple

xx swap

xx taca, help\_Mul\_series

辅助函数中使用了递归的方法。首先使用dup函数复制栈顶数值，one?, false?branch,2进行判断 ，为了不改变原来的序列，使用xx|over|x函数调整数列，在让末尾两数相乘合并，再swap末尾二数，为递归做准备。直到one?, false?branch,2进行判断结束，end。返回到Mul\_series函数中，此时末尾的数值是判断成功后的1，所以最后用drop把栈顶1去掉，完成并结束该函数运行 。

效果如下：



