

实验报告

PA1x

1. stack.cl 代码实现

构造 `StackCommand` 类。包含以下内容：

成员变量

```
class StackCommand inherits A2I
{
  stack : String;
  io : IO <- new IO;
```

`stack` :作为栈，类型为 `String` 。栈顶对应该字符串的最后一个字符；
`io` :IO类型实例，用于调用 `out_string()` 方法；
此外，该类还继承 `A2I` 类，以调用相关方法。

push() 方法

```
push(char : String) : SELF_TYPE
{
  {
    stack <- stack.concat(char);
    self;
  }
};
```

将输入的字符(字符串)连接到原字符串(加到栈)上。

display() 方法

```
display(): SELF_TYPE
{
  {
    let n : Int <- stack.length() - 1 in
    (
      while 0 <= n loop
      {
        io.out_string(stack.substr(n,1));
        io.out_string("\n");
        n <- n - 1;
      } pool
    );
    self;
  }
};
```

将字符串倒序按行输出。

evaluate() 方法

```
evaluate(): SELF_TYPE
{
  {
    let n: Int <- stack.length() in
    (
      if n = 0 then {self;}
      else if stack.substr(n - 1, 1) = "+" then
      {
        let x : Int <- a2i(stack.substr(n - 3, 1)), y: Int <- a2i(stack.substr(n - 2, 1)) in
        (
          let ans : String <- i2a(x + y) in
          stack <- stack.substr(0, n - 3).concat(ans)
        );
      }
      else if stack.substr(n - 1, 1) = "s" then
      {
        let x: String <- stack.substr(n - 3, 1), y: String <- stack.substr(n - 2, 1) in
        (
          let ans : String <- y.concat(x) in
          stack <- stack.substr(0, n - 3).concat(ans)
        );
      }
      else {self;}
    fi fi fi
    );
    self;
  }
};
```

分为四种情况：

1. **栈空**。此时什么都不做。
2. **栈顶是+号**。此时将倒数第三个字符和倒数第二个字符通过 `a2i()` 方法转换为整数，然后将二者相加，得到的和再通过 `i2a()` 方法转换回字符串，并与栈去除掉栈顶的三个元素的字符串相连，返回。
3. **栈顶是s**。此时将倒数第三个字符和倒数第二个字符通过 `concat()` 方法进行交换，然后与栈去除掉栈顶的三个元素的字符串相连，返回。
4. **其他情况**。由于push进栈的字符只有+号、s和整数三种类型，所以该情况对应栈顶是整数的情况，什么都不用做。

在 `Main` 类中，处理输入。

Main 类构造

```
class Main inherits IO
{
  sc : StackCommand <- new StackCommand;
  isStop : Bool <- false;
  main() : Object
  {
    while not isStop loop
    {
      out_string(">");
      let ch : String <- in_string() in
      (
        if ch = "e" then sc.evaluate()
        else if ch = "x" then isStop <- true
        else if ch = "d" then sc.display()
        else sc.push(ch)
        fi fi fi
      );
    } pool
  };
};
```

`sc` : `StackCommand` 类的实例;

`isStop` : `Bool` 类型，判断是否应该停止。

在 `Main()` 函数中，进行循环：每次输出一个提示符 `>`，并要求一个输入 `ch`。根据 `ch` 决定相应的操作。当输入x时，令 `isStop = true`，从而终止循环。

2. 实验结果

在该PA目录下，输入 `coolc stack.cl atoi.cl` 和 `spim -file stack.s` 执行代码，再依次输入1、+、2、s、d、e、e、d、x，执行结果如图。

```
● breezer@ubuntu:~/CompilerPA/assignments/PA1$ coolc stack.cl atoi.cl
● breezer@ubuntu:~/CompilerPA/assignments/PA1$ spim -file stack.s
SPIM Version 6.5 of January 4, 2003
Copyright 1990-2003 by James R. Larus (larus@cs.wisc.edu).
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: ../lib/trap.handler
>1
>+
>2
>s
>d
s
2
+
1
>e
>e
>d
3
>x
COOL program successfully executed
```