LéPix

Ceci n’est pas le Photoshop

( ᐛ_nvarchar) (> ヽ- ヽ)
(熊猫)
(●﹏●)
(ゎ_ゎ)
Introduction

- Imperative programming language
- Swift-esque syntax
- Compiles to LLVM
- Floating point values and arrays to enable image manipulation
Implementation

Text → Scanner + Parser → Ast → Semantic Analyzer

Augmented AST → Code generation → LLVM IR

Standard Library
Syntax

- Syntax is reminiscent of the Swift programming language

  Function declaration    fun gcd( a : int , b : int ) : int { ... }

  Variable declaration    var a : int = 42;

  Array declaration    var arr : int[[2,2]] = [1,2,3,4];

  Control Flow    C-style for , while and if-else

  Array Access    arr[[1,2]]
Semantics

- Semantics are checked with a depth first traversal of the AST
- All identifiers are checked for validity
- Expressions, statements and declarations are type checked
- Static scoping and strong typing rules are used
- Implicit float to int casting on assignment
Codegen

- Codegen is performed on a semantically checked, augmented AST
- Symbol table holds pointers to global and local variables (including arrays)
- Functions definitions are held in a function table
- 2-D matrices are allocated as 1-D arrays and indexed accordingly
- Standard library is automatically included at compile time
- Stdlib:
  - Math : Modulo, Square Root, Prime Sieve, GCD etc.
  - I/O : Printing to console for different types (Int, Bool, Float), Printing an RGB image in PPM format
Testing: Building Regression Test Suite

- Began with simple examples based on how we wanted our language to look/behave (e.g. hello world)
- Based first set of tests on MicroC
- Revised test suite whenever language syntax/behavior changed
- Expanded test suite whenever new functionality was added to language (e.g. 1D array and 2D array)
- Coverage: tested for both expected behavior and potential user error when applicable
Testing: Continuous Integration with Travis
Testing: When Everything Works

The command "./testall.sh" exited with 0.
Testing: When Everything Breaks (ಥ﹏ಥ) (ಥ﹏ಥ)

The command "./testall.sh" exited with 1.

Done. Your build exited with 1.
Example LePix programs

```java
1  fun mod(num:int, denom:int): int {
2      var multiplier:int = num/denom;
3      return num - denom * multiplier;
4  }
5  
6  fun isPrime(test:int): bool {
7      var i:int;
8      var ans:bool = true;
9      for(i=2; i < test; i = i + 1) {
10         if(mod(test, i) == 0) {
11             ans = false;
12         }
13      }
14      return ans;
15  }

16  fun main(): int {
17      printb(isPrime(5));
18      printb(isPrime(9));
19      printb(isPrime(46));
20      printb(isPrime(57));
21      printb(isPrime(12));
22      printb(isPrime(103));
23      printb(isPrime(96));
24      return 0;
25  }
```

```java
1  fun main(): int
2  {
3      var img: int[15552] = [229, 236, 239, 229, 240, 240,
4          w: int = 72;
5          h: int = 72;
6          var i: int;
7          var temp: int;
8          var avg: int;
9          for(i = 0; i < 15550; i = i + 3)
10             {
11                 temp = img[i] + img[i+1] + img[i+2];
12                 avg = temp/3;
13                 img[i]=avg;
14                 img[i+1]=avg;
15                 img[i+2]=avg;
16             }
17      printppm(w);
18      var j: int;
19      for (j=0; j<15552; j=j+1){
20             print(img[j]);
21         }
22      return 0;
23  }
```
Images
fun demo(): void {}