Assignment 4 - CS335

Group 14

April 4, 2016

1 Build and Run

- `cd asgn4`
- `make`
- `bin/irgen file`

2 Features

- **Declaration** : `val` and `var` declaration are supported. Also, It is necessary that declaration be accompanied by a value. It is optional to specify type. Multiple variables can be declared in same line.

```
val a = 5;
val b : Int = 27,  aldo = 21;
var c = 312;
```

- **Array** : Integer Arrays are Supported with predefined length. They can be used just like any other other variable with standard array referencing.

```
val c = new Array[Int](21);
c[5] = a;
a = c[20]*2;
```

- **Objects** : There can be multiple singleton objects(Scala-Like). They cannot be referenced from each other. Also, code in outer most scope of both the objects will be executed(unlike scala, in which only the code of objects extending `App` are executed.)

```
object HelloWorld {

}
```

- **For** : Nested For loops are implemented with new scope beginning at each loop. There are two variants `to` and `until`

```
for ( i <- 23 to 71) {
    val j = 32;
    for ( j <- 21 until 23) a = a*2;
    print();
}
```

- **While** : Nested While loops are implemented with new scope beginning at each loop.

```
while(a >= 2) {
     print();
     a = a - 1;
     val b  = 32;
     while( b < 50) {
            b = b + 1 ;
       }
}
```

- **If/Else** : Nested If/Else are implemented with new scope at each if, else.

```
if(a ==31) {
   print();
   if(b == 5) {
       print();
   }
}
else {
   a = 31;
}
```

- **Case Switch** : Case/switch are supported and no new scope is made for them. We have not allowed fall-through and have clubbed all the conditional statements together for more efficient n-way branch (due to cache-hits while fetching instructions)

```
2*c[20] + 1 match{
  case b * 2 => a = 2;
  case 7 => a = 3;
  case 2 => {a = 4; a = 6;}
}
```

- **Functions** : Functions with multiple arguments and Single return value. A new scope is formed for each function. Recursion is also suported.

```
 def print() = {
    val a = 2;
    print();
    return a;
}
```