

Appendix A

The Function class

Line-break appropriately

```
1  import java.util.ArrayList;
2  import java.util.function.BiFunction;
3
4  public class Function extends HeapObject {
5      private BiFunction<HeapObject[], HeapObject[], HeapObject> inner;
6      private HeapObject[] freeVariables;
7      private ArrayList<HeapObject> arguments;
8      private int arity = 0;
9      private HeapObject result = null;
10
11     public Function(BiFunction<HeapObject[], HeapObject[], HeapObject> inner, int arity, HeapObject[] freeVariables) {
12         this.inner = inner;
13         this.arity = arity;
14         this.freeVariables = freeVariables;
15         arguments = new ArrayList<>();
16     }
17
18     @Override
19     public HeapObject enter() {
20         // Check if we've got a cached value
21         if (result != null) {
22             return result;
23         }
24
25         if (arguments.size() < arity) {
26             return this;
27         }
28         else if (arguments.size() > arity) {
29             try {
30                 Function fun = (Function)inner
31                     .apply(arguments.subList(0, arity).toArray(new HeapObject[0]), freeVariables)
32                     .enter()
33                     .clone();
34                 for (HeapObject arg : arguments.subList(arity, arguments.size()))
```

```

35         fun.addArgument(arg);
36         result = fun.enter();
37         return result;
38     }
39     catch (CloneNotSupportedException e) {
40         throw new RuntimeException(e);
41     }
42 }
43 else {
44     result = inner.apply(arguments.toArray(new HeapObject[0]), freeVariables).enter();
45     return result;
46 }
47 }
48
49 public void addArgument(HeapObject arg) {
50     arguments.add(arg);
51 }
52
53 @Override
54 public Object clone() throws CloneNotSupportedException {
55     Function f = (Function)super.clone();
56     f.inner = inner;
57     f.arity = arity;
58     f.freeVariables = freeVariables.clone();
59     f.arguments = new ArrayList<>(arguments);
60     return f;
61 }
62 }

```
