

Appendix A

The `Function` class

```
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,
12         ↪ HeapObject[] freeVariables) {
13         this.inner = inner;
14         this.arity = arity;
15         this.freeVariables = freeVariables;
16         arguments = new ArrayList<>();
17     }
18
19     @Override
20     public HeapObject enter() {
21         // Check if we've got a cached value
22         if (result != null) {
23             return result;
24         }
25
26         if (arguments.size() < arity) {
27             return this;
28         }
29         else if (arguments.size() > arity) {
30             try {
31                 Function fun = (Function)inner
32                     .apply(arguments.subList(0, arity).toArray(new HeapObject[0]),
33                     ↪ freeVariables)
34                     .enter()
35                     .clone();
36                 for (HeapObject arg : arguments.subList(arity, arguments.size()))
37                     fun.addArgument(arg);
38                 result = fun.enter();
39                 return result;
40             }
41             catch (CloneNotSupportedException e) {
42                 throw new RuntimeException(e);
43             }
44         }
45     }
46 }
```

```

44         result = inner.apply(arguments.toArray(new HeapObject[0]),
45                               ↪ freeVariables).enter();
46         return result;
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 }

```
