Programming Languages

First Phase Report - Lazy Lists

José António Ribeiro da Silva Lopes
ist1103938

Big Step Evaluation rules

$$\frac{}{\mathcal{E}; \text{lcons}(N, M) \Downarrow \text{lcons}(N, M, \mathcal{E}); S}$$

$$\frac{\mathcal{E}; S; M \Downarrow \text{nil}; S' \quad \mathcal{E}; S' N \Downarrow U; S''}{\mathcal{E}; S; \text{match } M\{ \mid \text{nil} \to N \mid y :: z \to K\} \Downarrow U; S''}$$

$$\frac{\mathcal{E}; S; M \Downarrow \text{lcons}(I, J, \mathcal{E}); S' \quad I \Downarrow R \quad J \Downarrow T \quad \mathcal{E}[y \to R; z \to T]; S'; K \Downarrow U; S''}{\mathcal{E}; S; \text{match } M\{ \mid \text{nil} \to N \mid y :: z \to K\} \Downarrow U; S''}$$

Implementation

When evaluating an AST node corresponding to a lazy list, the values of both `M` and `N` are only saved as other AST nodes, meaning they have yet to be evaluated. Once they are, via the match construct, they become proper IValues and a flags flips to true in order to indicate that they indeed have been evaluated. All of this information is saved in a VLCons IValue.

The structure of an ASTLCons node is, then, the following:

```
public class ASTLCons implements ASTNode {
  ASTNode head, tail;

  public ASTLCons(ASTNode head, ASTNode tail) {
    this.head = head;
    this.tail = tail;
  }

  public IValue eval(Environment<IValue> e) throws InterpreterError {
    return new VLCons(this.head, this.tail, e);
  }
}
```

And the structure of VLCons is as follows:

```
public class VLCons implements IValue {
  ASTNode head, tail;
  Environment<IValue> e;

  IValue evaluatedHead = null;
  IValue evaluatedTail = null;
  boolean headWasEvaluated = false;
  boolean tailWasEvaluated = false;

  public VLCons(ASTNode head, ASTNode tail, Environment<IValue> e) {
    this.head = head;
    this.tail = tail;
```

```
    this.e = e;
  }
  ...
}
```

The magic happens inside the match node, in the case where it matches a non nil list, because then, `M` and `N` are evaluated. Like so:

```
// VLCons.java
...
public IValue getHead() throws InterpreterError {
  if (!this.headWasEvaluated) {
    this.evaluatedHead = this.head.eval(this.e);
    this.headWasEvaluated = true;
  }
  return this.evaluatedHead;
}

public IValue getTail() throws InterpreterError {
  if (!this.tailWasEvaluated) {
    this.evaluatedTail = this.tail.eval(this.e);
    this.tailWasEvaluated = true;
  }
  return this.evaluatedTail;
}
...
// ASTMatch.java
...
VLCons lc1 = (VLCons) v1;

IValue v2 = lc1.getHead();
IValue v3 = lc1.getTail();
Environment<IValue> en = new Environment<IValue>(e);
en.assoc(headName, v2);
en.assoc(tailName, v3);

return consCase.eval(en);
...
```