Big-Step Evaluation Store:

$$\frac{}{E\,;nil \Downarrow nil}$$

$$\frac{}{E\,;I\,cons(M,N) \Downarrow I\,cons(M,N,En)\,;S}$$

$$\frac{E\,;S\,;M \Downarrow nil\,;S^{'}\,E\,;S^{'}\,N \Downarrow U\,;S^{''}}{E\,;S\,;match\,M\,\{nil \rightarrow N \vee cons(x,l) \rightarrow R\} \Downarrow U\,;S^{''}}$$

$$\frac{E\,;S\,;M \Downarrow Icons(V,L,F)\,;S^{'}F\,;S^{'}\,;V \Downarrow J\,;S^{''}F\,;S^{''}\,;L \Downarrow K\,;S^{'''}E[x \rightarrow J][l \rightarrow K];S^{'''};R \Downarrow U\,;S^{''''}}{E\,;S\,;match\,M\,\{nil \rightarrow N \vee cons(x,l) \rightarrow R\} \Downarrow U\,;S^{''''}}$$

Implementation:

From an implementation stand point. The core difference lies in **when** the elements of the list are evaluated:

On the creation of the List class i evualuated both Nodes that i received:

```
return new Vcons(head.eval(e), tail.eval(e));
```

While on Lazy List i did not but to account for the later evaluation i added an enviroment:

```
return new Vicons(head, tail, e);
```

On match command now the Lazy list needs to be evaluated while List does not:

```
newEnv.assoc(headIdentifier, iconsValue.getHead().eval(iconsValue.getEnv()));

newEnv.assoc(tailIdentifier, iconsValue.getTail().eval(iconsValue.getEnv()));
```

vs

```
newEnv.assoc(headIdentifier, iconsValue.getHead());

newEnv.assoc(tailIdentifier, iconsValue.getTail());
```