Exercise 4.28.

`Eval` uses `actual-value` rather than `eval` to evaluate the operator before passing it to `apply`, in order to force the value of the operator. Give an example that demonstrates the need for this forcing.

Answer.

Consider the following definition of `average-damp` in section 1.3.4:

```
(define (average-damp f)
  (lambda (x) (average x (f x))))
```

If we remain using `eval` to evaluate the operator before passing it to `apply`, the evaluator would fail to dispatch on type of the operator, which is a thunk. Applying this resulting procedure to 10 will cause the evaluator run into bewilderment:

```
((average-damp square) 10)
;Unknown procedure type - APPLY (thunk square (...))
```

For the argument `square` has been packaged into a thunk, say, `('thunk square the-global-env)`, and can not be recognized by `eval`. Only then `eval` adopt `actual-value` to force the value of the thunk does it correctly get `square` applied.

```
((average-damp square) 10)
55
```


Email address: informlarry@gmail.com