Exercise 4.70.

What is the purpose of the `let` bindings in the procedures `add-assertion!` and `add-rule!`? What would be wrong with the following implementation of `add-assertion!`? Hint: Recall the definition of the infinite stream of ones in section 3.5.2: `(define ones (cons-stream 1 ones))`.

```
(define (add-assertion! assertion)
  (store-assertion-in-index assertion)
  (set! THE-ASSERTIONS
        (cons-stream assertion THE-ASSERTIONS))
  'ok)
```

Answer.

The `let` bindings in the procedures `add-assertion!` and `add-rule!` forces `THE-ASSERTIONS` to reveal the complete stream of assertions or rules so that it points to the stream of one more assertion or rule after the assignment. Recall that assertions and rules are stored in delayed streams and their `cdr` won't be evaluated unless requested. In the new implementation, `THE-ASSERTION` always points to streams with their `cdr` delayed throught the assignment. So instead of augmenting the assertion stream, the assignment expression replace the `car` of the stream with the designated assertion.

Email address: informlarry@gmail.com