Exercise 4.46.

The evaluators in sections 4.1 and 4.2 do not determine what order operands are evaluated in. We will see that the amb evaluator evaluates them from left to right. Explain why our parsing program wouldn't work if the operands were evaluated in some other order.

Answer.

When it comes to complex grammars, our parsing program performs searching and matching among possibilities of how the units can be decomposed. Specifically, both `parse-verb-phrase` and `parse-noun-phrase` exploits `amb` to check the part of speech of the unparsed phrase. Their internal procedures `maybe-extend` are implemented in a self-recursive way.

If the operands were evaluated from right to left for example, our parsing program would stuck in an infinite recursion when encountered a sentence contains proposition. Consider the sentence "The cat sleeps in the class."

```
(parse '(the cat sleeps in the class))

(set! *unparsed* '(the cat sleeps in the class))

(parse-sentence)

(list 'sentence
      (parse-noun-phrase)
      (parse-verb-phrase))

(list 'sentence
      (maybe-extend (parse-simple-noun-phrase))
      (maybe-extend (parse-word '(verb studies lectures eats sleeps))))

(list 'sentence
      (maybe-extend
       (list 'simple-noun-phrase
             (parse-word '(article the a))
             (parse-word '(noun student professor cat class))))
      (maybe-extend
       (parse-word '(verb studies lectures eats sleeps))))

(list 'sentence
      (amb (list 'simple-noun-phrase
                 (parse-word '(article the a))
                 (parse-word '(noun student professor cat class)))
           (maybe-extend
            (list 'noun-phrase
                  (maybe-extend
                   (list 'simple-noun-phrase
                         (parse-word '(article the a))
                         (parse-word '(noun student professor cat class))))
                  (parse-prepositional-phrase))))
      (maybe-extend
       (parse-word '(verb studies lectures eats sleeps))))

...
```

We can see that the second argument in the expression `(amb <arg1> <arg2>)` is expanding eternally. And the process generated above indicates that only when the arguments to `amb` were evaluated form left to right did our parsing program work correctly.