Exercise 4.39.

Does the order of the restrictions in the multiple-dwelling procedure affect the answer? Does it affect the time to find an answer? If you think it matters, demonstrate a faster program obtained from the given one by reordering the restrictions. If you think it does not matter, argue your case.

Answer.

Arranging the restrictions in different order does not affect the answer, for all of them must be met any way, thereby confining the answer in the same set. However, the order of restriction does have great impact on the performance of the program.

In nondeterministic evaluation, exploration is carried out among branches using choronological backtracking. If we can eliminate impossible choices as much as possible at the very begin, then the performance of the program will be greatly improved. Among these restrictions, the one "Miller lives on a higher floor than does Cooper." eliminates choice of dead ends most intensively (almost a half). Hence, it should be placed at the beginning among these requirements. Besides, process generated by the predicate `distinct?` grows as $\Theta\left(\phi^n\right)$, where ϕ is the golden ratio described in section 1.2.2. So running it over the reduced cases rather than the raw input would also increase the efficiency of the evaluator. Using these ideas, we can modify the `multiple-dwelling` procedure in the text as following, which is expected to run much faster:

```
(define (multiple-dwelling)
  (let ((baker (amb 1 2 3 4 5))
        (cooper (amb 1 2 3 4 5))
        (fletcher (amb 1 2 3 4 5))
        (miller (amb 1 2 3 4 5))
        (smith (amb 1 2 3 4 5)))
    (require (> miller cooper))
    (require (not (= baker 5)))
    (require (not (= cooper 1)))
    (require (not (= fletcher 5)))
    (require (not (= fletcher 1)))
    (require (not (= (abs (- smith fletcher)) 1)))
    (require (not (= (abs (- fletcher cooper)) 1)))
    (require
     (distinct? (list baker cooper fletcher miller smith)))
    (list (list 'baker baker)
          (list 'cooper cooper)
          (list 'fletcher fletcher)
          (list 'miller miller)
          (list 'smith smith))))
```

Email address: informlarry@gmail.com