Assignment 5 Solutions

Nathan Hitchings

January 17, 2019

1 Exercise 1.20

The process that a procedure generates is of course dependent on the rules used by the interpreter. As an example, consider the iterative gcd procedure given above. Suppose we were to interpret this procedure using normal-order evaluation, as discussed in Section 1.1.5. (The normal-order-evaluation rule for if is described in Exercise 1.5.) Using the substitution method (for normal order), illustrate the process generated in evaluating (gcd 206 40) and indicate the remainder operations that are actually performed. How many remainder operations are actually performed in the normal-order evaluation of (gcd 206 40)? In the applicative-order evaluation?

1.1 Solution

This is the gcd procedure:

```
(define (gcd a b)
  (if (= b 0)
      a
      (gcd b (remainder a b))))
```

Evaluating (gcd 206 40) using normal order we get

```
;; Normal order
(gcd 206 40)

(if (= 40 0)
    206
    (gcd 40 (remainder 206 40)))

;; expand gcd
```

```
(if (= (remainder 206 40) 0)
    40
    (gcd (remainder 206 40) (remainder 40 (remainder 206 40))))
(if (= 6 0) ; 1
    40
    (gcd (remainder 206 40) (remainder 40 (remainder 206 40))))

;; expand gcd
(if (= (remainder 40 (remainder 206 40)) 0)
    (remainder 206 40)
    (gcd (remainder 40 (remainder 206 40))
         (remainder (remainder 206 40)
                    (remainder 40 (remainder 206 40)))))
(if (= (remainder 40 6) 0) ; 2
    (remainder 206 40)
    (gcd (remainder 40 (remainder 206 40))
         (remainder (remainder 206 40)
                    (remainder 40 (remainder 206 40)))))
(if (= 4 0) ; 3
    (remainder 206 40)
    (gcd (remainder 40 (remainder 206 40))
         (remainder (remainder 206 40)
                    (remainder 40 (remainder 206 40)))))
;; expand gcd
(if (= (remainder (remainder 206 40) (remainder 40 (remainder 206 40))) 0)
    (remainder 40 (remainder 206 40))
    (gcd (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
         (remainder (remainder 40 (remainder 206 40))
                    (remainder (remainder 206 40)
                               (remainder 40 (remainder 206 40))))))
(if (= (remainder 6 (remainder 40 6)) 0) ; 4 and 5
    (remainder 40 (remainder 206 40))
    (gcd (remainder (remainder 206 40) (remainder 40 (remainder 206 40)))
         (remainder (remainder 40 (remainder 206 40))
                    (remainder (remainder 206 40)
                               (remainder 40 (remainder 206 40))))))


;; Applicative order
(gcd 206 40)
```

```
(gcd 40 (remainder 206 40))
(gcd 40 6)
(gcd 6 (remainder 40 6))
(gcd 6 4)
(gcd 4 (remainder 6 4))
(gcd 4 2)
(gcd 2 (remainder 4 2))
(gcd 2 0)
2
```

So, `remainder` is evaluated 4 times using applicative order and a lot more times (18) using normal order.