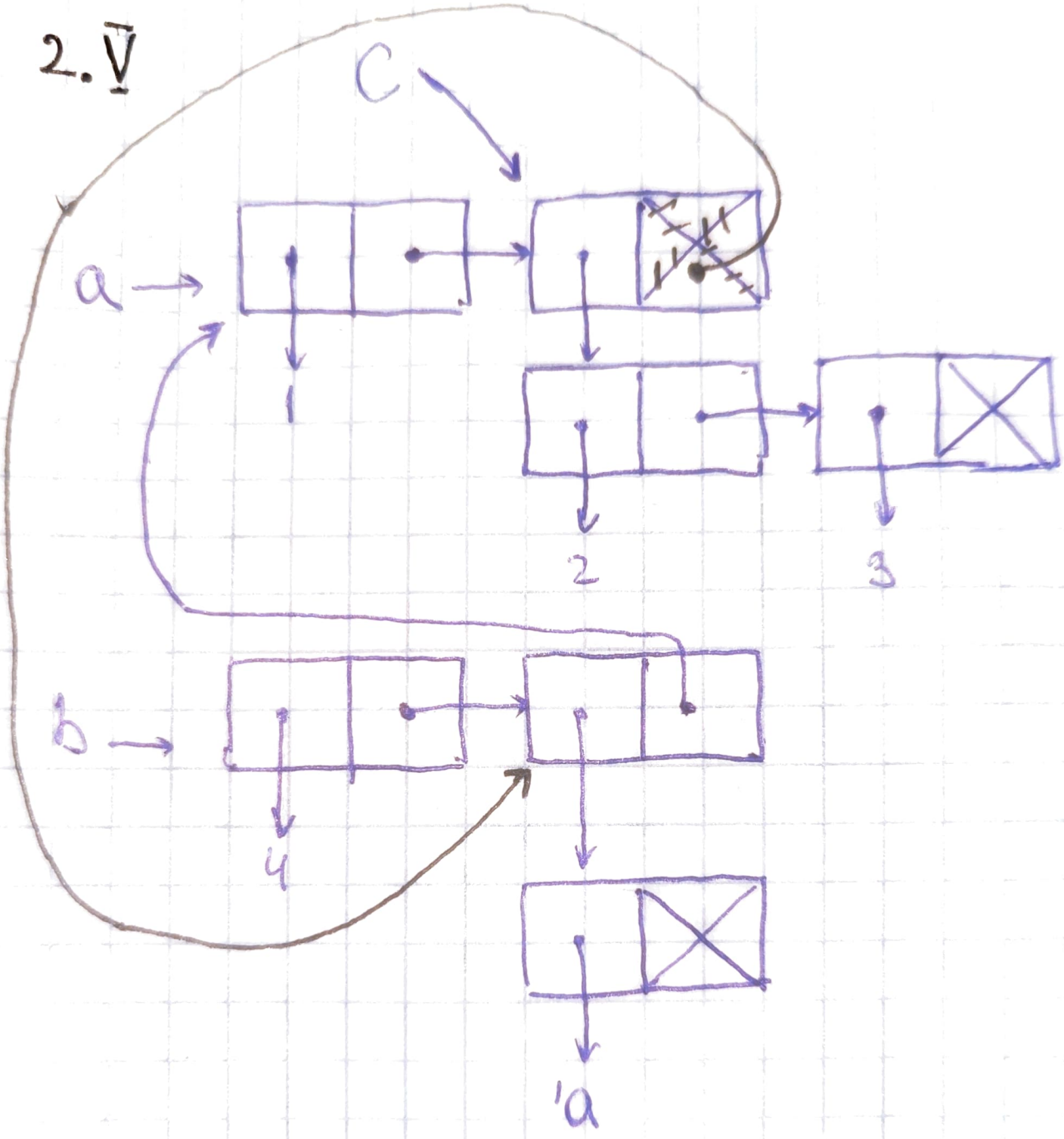
```
#lang racket/base
(require racket/stream)
; 2.I

(define-syntax when
    (syntax-rules ()
        ((_ test)
            (if
                test
                test
                #f
            )
        )
        ((_ test expr1)
            (if
                test
                expr1
                #f
            )
        )
        ((_ test expr1 expr2 ...)
            (if
                test
                (begin expr1 (when #t expr2 ...))
                #f
            )
        )
    )
)

; 2.II

(define (filter1 f lst)
    (reverse
        (foldl
            (lambda (x y) (if (f x) (cons x y) y))
            null
            lst
        )
    )
)
(define (filter2 f lst)
    (foldr
        (lambda (x y) (if (f x) (cons x y) y))
        null
        lst
    )
)

; 2.III
; (λz. (λx. ((λy. (x z)) ((λy. y y) (λy. y y z))))) a b  -> (по alpha)
; (λz. (λx. ((λy. (x z)) ((λw. w w) (λv. v v z))))) a b  -> (по Beta)
; (λx. ((λy. (x a)) ((λw. w w) (λv. v v a)))) b  -> (по Beta)
; (λy. (b a)) ((λw. w w) (λv. v v a)) -> (по Beta)
```

```
; (b a) == Нормальная форма, далее невозможны Beta-редукции

; 2.IV
(define (nthbit n)
    (let loop ((x n))
        (if (= x 1)
            1
            (let ((del (remainder x 2)))
                (if (= del 1)
                    0
                    (loop (/ x 2))
                )
            )
        )
    )
)
```

2. V



```
; 2.VI
(define (div3 n)
    (if (> (remainder n 3) 0)
        n
        (div3 (/ n 3))
    )
)

(define (div5 n)
    (if (> (remainder n 5) 0)
        n
        (div5 (/ n 5))
    )
)


(define (power35? n)
    (if (= (div5 (div3 n)) 1)
        #t
        #f
    )
)

(define (ints-from n)
    (stream-cons n (ints-from (+ n 1))))
(define ints (ints-from 1))

(define (stream3^m5^n-from ints)
    (let ((first (stream-first ints)))
        (if (power35? first)
            (stream-cons first (stream3^m5^n-from (stream-rest ints)))
            (stream3^m5^n-from (stream-rest ints))
        )
    )
)

(define stream3^m5^n (stream3^m5^n-from ints))
```