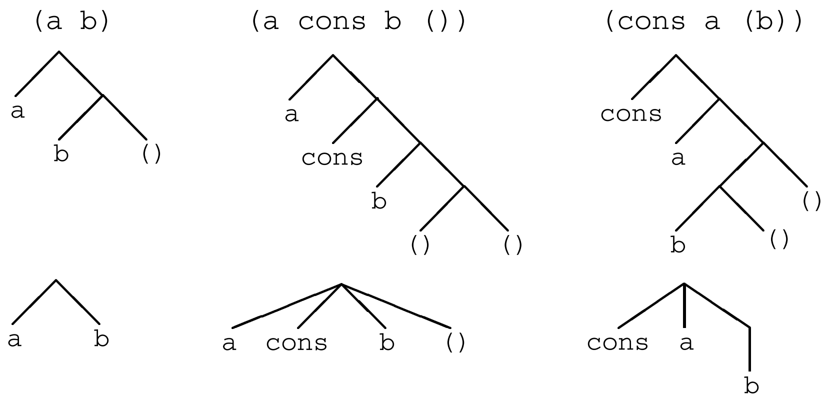
Musterlösung VI. Übungsblatt "Informatik I" WS 95/96

Jörg Fischer, Christian Schulte

Aufgabe VI.1 (Listen) Die Ausdrücke evaluieren zu den folgenden Werten:

1. `(a b)`
2. `(a b)`
3. `(a b)`
4. `(a cons b ())`
5. `(a b)`
6. `(a b)`
7. `(cons a (b))`

Die dazugehörigen Binär- und L-bäume:



Aufgabe VI.2 (Filtern von Listen)

```
(define (filter bed? xs)
  (cond ((null? xs)
         ())
        ((bed? (car xs))
         (cons (car xs) (filter bed? (cdr xs))))
        (else
         (filter bed? (cdr xs)))))
```

Aufgabe VI.3 (Komprimieren von Listen)

1. Maximal lange RLE-Kodierungen entstehen für:

$$(p_0 ... p_n) \qquad \text{mit:} \quad p_j \neq p_{j+1} \forall 0 \leq j < n$$

   Minimal lange RLE-Kodierungen für:

$$(p_0 ... p_n) \qquad \text{mit:} \quad p_j = p_{j+1} \forall 0 \leq j < n$$

2.
```
(define (run ys n x)
  (cond ((null? ys)
         (list (cons n x)))
        ((eq? x (car ys))
         (run (cdr ys) (+ n 1) x))
        (else
         (cons (cons n x) (run (cdr ys) 1 (car ys))))))

(define (rle xs)
  (if (null? xs)
      ()
      (run (cdr xs) 1 (car xs))))
```

3.
```
(define (expand n x)
  (if (= n 0)
      ()
      (cons x (expand (- n 1) x))))

(define (unrle xs)
  (if (null? xs)
      ()
      (append (expand (caar xs) (cdar xs)) (unrle (cdr xs)))))
```

Aufgabe VI.4 (Umdrehen von Listen)

```
(define (rev-help xs ys)
  (if (null? xs)
      ys
      (rev-help (cdr xs) (cons (car xs) ys))))

(define (reverse xs)
  (rev-help xs ()))
```

Aufgabe VI.5 (Blätter eines Baumes)

```
(define (blaetter xs)
  (cond ((pair? xs)
         (append (blaetter (car xs)) (blaetter (cdr xs))))
        ((null? xs)
         xs)
        (else
         (list xs))))
```

Aufgabe VI.6 (Gleichheit von Listen)

```
(define (gleich? xs ys)
  (if (and (pair? xs) (pair? ys))
      (and (gleich? (car xs) (car ys))
           (gleich? (cdr xs) (cdr ys)))
      (eq? xs ys)))
```