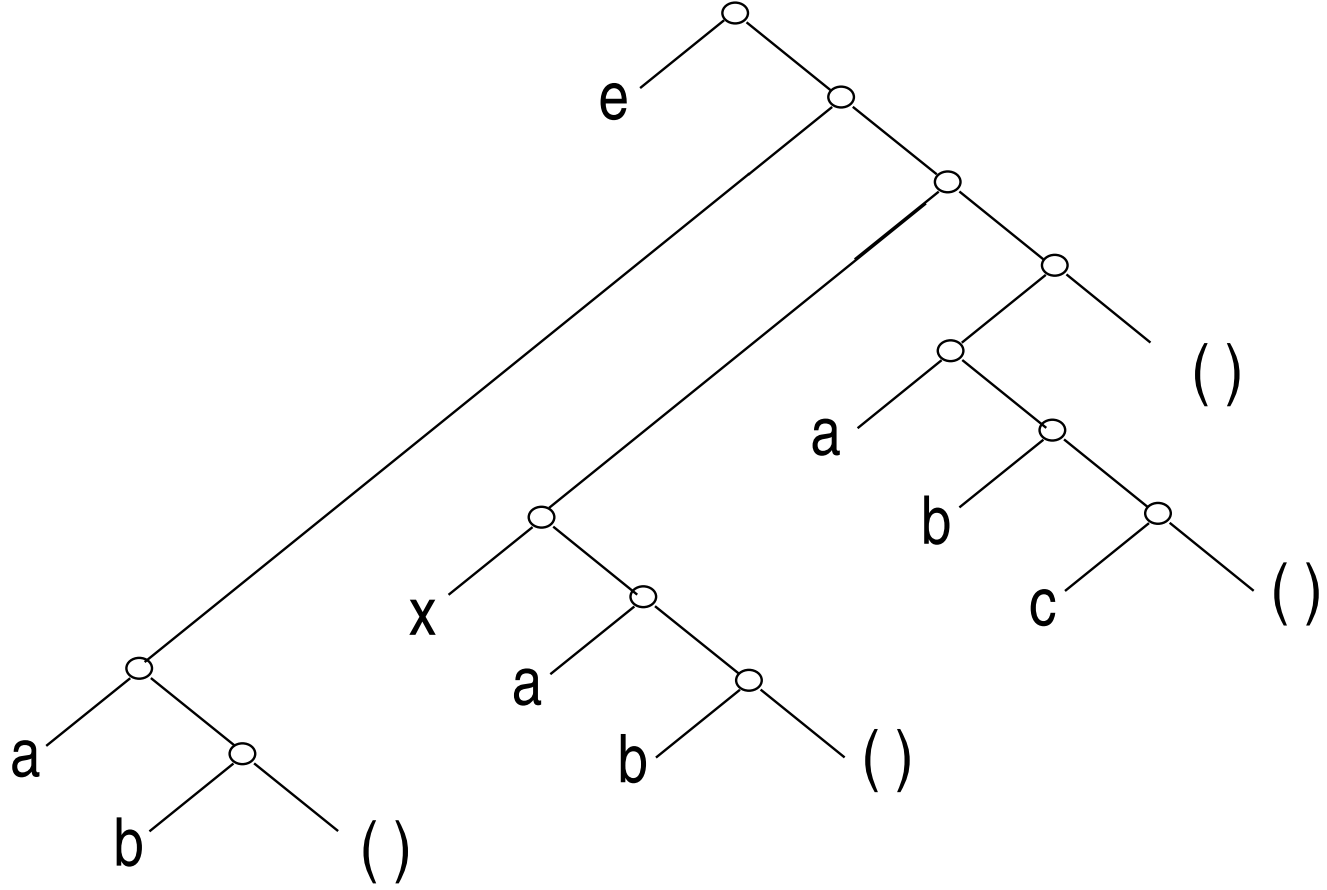


```
(define subseff  (lambda (x y z)
    (cond ((equal? y z)  x)
          ((not (pair? z))  z)
          (#t  (consifnew z
                          (subseff x y (car z))
                          (subseff x y (cdr z)) )) ) ))
```

```
(define consifnew  (lambda (z newcar newcdr)
    (cond ((and (eq? (car z) newcar)
                (eq? (cdr z) newcdr) )
           z)
          (#t  (cons newcar newcdr)) ) ))
```