CvP - Werkcollege 7

Exercise 1 Consider the λ-term $M = (\lambda x.xy)(\lambda z.z(xy))$.

(a) Compute the free variables free(M) of M.

(b) Use an α-conversion to ensure that every occurrence of x in M is free.

(c) Use β-reductions to compute a beta normal form of M.

Exercise 2 A λ-term possibly reduces in multiple different ways. The *Church-Rosser property* states that each reduction can lead to the same λ-term: If $\to^*$ is the transitive closure of $\to$, $M \to^* A$ and $M \to^* B$, then there exists some λ-term N, such that $A \to^* N$ and $B \to^* N$.

Consider the λ-term $M = (\lambda x.((\lambda y.yx)z))w$.

(a) Give two different β-reductions of M.

(b) Show that both reductions lead to same beta normal form.

Exercise 3 The *Church-Rosser property* ensures that the beta normal form is unique. We can, therefore, speak about *the* beta normal form. However, the beta normal need not necessarily exist.

Consider the λ-term $\omega = \lambda x.xx$. Show that $\Omega = \omega\omega$ has no beta normal form.

Exercise 4 Consider the λ-term $Y = \lambda g.(\lambda x.g(xx))(\lambda x.g(xx))$, which is called the *fixed-point combinator*. Show that YF and $F(YF)$ both reduce to the same λ-term.

Exercise 5 What do the following Scheme functions do?

(a)
```
(define (y s lis)
  (cond
    ((null? lis) '() )
    ((equal? s (car lis)) lis)
    (else (y s (cdr lis)))))
```

(b)
```
(define (x lis)
  (cond
    ((null? lis) 0)
    ((not (list? (car lis)))
      (cond
        ((eq? (car lis) #f) (x (cdr lis)))
        (else (+ 1 (x (cdr lis))))))
    (else (+ (x (car lis)) (x (cdr lis))))))
```