Tarea 5.

Leonardo Brambila Ayala.

Lenguajes de programación
Lic. en Ciencias de la Computación
Universidad de Sonora
Semestre 2023-1

1 Descripcion de las las funciones.

(define (read-char* input)
(define ch (read-char input))
(when debug?
(printf "read char s from input ch)
R = Lee un caracter del imput, mediante la funcion "read-char", y lo que se leyo se guardara en la variable local "ch". Despues verifica en debug y luego imprime un mensaje usando printf.

Seg.Fun

(define (peek-char* input)
(define ch (peek-char input))
(when debug? (printf "peeked char s from input ch)
R= La funcion peek-char*, se utiliza para obtener el proximo aracter de la fuente de entrada (input) sin consumirlo, y este caracter se guardara en el variable ch.

Ter.Fun

(define (char-digit? ch) (char<=? # \0 ch # \9))
R = ch: Es el caracter que se va a verificar.
La función utiliza la función char<=? para comparar el carácter ch con los caracteres 0-9. Devuelve verdadero si esta dentro del rango establecido si no devolvera falso.
Detrminina si el caracyer pasado como argumento es un digito decimal del 0-9. Su resultado de "char<=?" sera verdadero si el caracter es un dijito, si no sera falso

Cuar.Fun

(define (char-varletter? ch) (member ch '(#\x #\y #\z) char=?))
R = ch: Es el caracter que se va a verificar.
Verifica si el caracter a verificar esta dentro de la lista prmitida, "x, y, z. " La funcion devuelve el primer elemento de la lista cuando es igual a la variable "ch".
En conclusion verifica si el caracter "ch: es una de las letras dentro del rango, si es asi devolvera verdadero, si no devolvera falso.

Quin.Fun

(define (char-delimiter? ch) (or (eof-object? ch) (char-whitespace? ch) (member ch '(#\(#\)))))

R = ch: Es el caracter que se va a verificar.

Determina si es caracter, es un delimitador, esto puede inclui espacios en blancos y parevtesis, si es ese caso devolvera verdadero si no falso.

Sex.Fun

(define (token-open-paren) (token 'open-paren #f lex-line lex-col))

Crea una instanci en Toen con su propio tipo open-paren, el valor de falso y las posiciones de las columnas y sus lineas actuales

Sep.Fun

(define (token-close-paren) (token 'close-paren #f lex-line lex-col))

Crea una instanci en Toen con su propio tipo clsose-paren, el valor de falso y las posiciones de las columnas y sus lineas actuales

Opt.Fun

(define (token-binop type) (token 'binop type lex-line lex-col))

Toma un argumento y crea una nueva instancia de Token con el tipo binop, el valor del argumento y las posiciones de las columnas y lieas actuales

Novena.Fun

(define (token-number value col) (token 'number value lex-line col))

Toma dos argumentos, value y col, y crea una instancia de token con el tipo 'number, el valor value, y las posiciones de las columnas y lineas actuales.

Decima.Fun

(define (token-number/+ token)
(token 'number
(token-value token)
(token-line token)
(sub1 (token-col token))))

Toma un token como argumento. Crea un nuevo token , el valor del token original (token-value token), la misma línea, pero con la columna decrementada en 1. La operación esta relacionada con la adición (+), ya que la columna se decrementa en 1.

Onceava.Fun

(define (token-number/- token)
(token 'number (- (token-value token)) (token-line token)
(sub1 (token-col token))))
Crea un nuevo token con el tipo 'number, con el valor negativo del token, en la misma línea (token-line token), pero con la columna decrementada en 1.

2 Errores y o bugs del codigo

1. En la funcion "(define(read-alphanumeric)..)"

2. En la funcion "(define (token-number/- token)...)"