

## TP 2: intérprete de TLC-Lisp codificado en CLisp

### Implementación

La función mas importante del intérprete es la función EVAL que recibe la expresión a evaluar y el ambiente en el que se encuentra el intérprete en ese momento:

( DEFUN EVAL ( EXPR AMB )...

El ambiente se trata como una lista de elementos con 2 componentes: el nombre y lo que ese nombre tenga asociado:

Ambiente: ( ( NOMBRE ASOCIACION ) ( NOMBRE ASOCIACION ) ... )

Entonces si el intérprete recibe un número devuelve lo mismo (por ser un número un átomo). En cambio si recibe un nombre debe revisar el ambiente para saber cual es su asociación (si es que ésta existe).

La función EVAL deberá llamarse usando quotes:

Ejemplo: ( EVAL ` ( CONS A B ) ` ( A X B ( Z Y ) ) )

Ese llamado devolvería la lista ( X Z Y ). Si no se utilizaran los quotes CLisp evaluaría las expresiones dentro de su ambiente, rompiendo así la “burbuja” impuesta al establecer un ambiente para la función EVAL.

### Código dado en clase

```
( DEFUN EVAL ( EXP AMB )  
  ( IF ( ATOM EXP ) ( IF ( NUMBERP EXP ) EXP ( BUSCAREXPENAMB ) )  
    ( COND ( ( EQ ( CAR EXP ) `QUOTE ( CADR EXP ) )  
              ( ( EQ ( CAR EXP ) `IF ) ( IF ( EVAL ( CADR EXP ) AMB )  
                                              ( EVAL ( NTH 2 EXP ) AMB )  
                                              ( EVAL ( NTH 3 EXP ) AMB )  
...  
...  
...  
...  
Así como se ve la implementación del IF quedan por hacer las implementaciones de COND, AND,  
OR y LAMBDA: ( EQ ( CAR EXP ) `LAMBDA ) EXP )
```

Evaluación de funciones que evalúan sus parámetros

En el caso de las funciones que evalúan sus parámetros el procedimiento a seguir es el mismo que el que utilizan las funciones:

1. Evaluar argumentos
2. Aplicar la función

Ejemplo: ( UNION ` ( a b c ) ( INTERSEC B C ) ): primero se evalúa el parámetro (se ejecuta INTERSEC B C) y luego se hace la unión con ( a b c ) (que no necesita ser evaluado)

```
...  
...  
...  
      ( T ( APLICAR ( CAR EXP ) (...) (...) )  
...  
...  
...
```

La función APLICAR tiene como parámetros la función, la lista de argumentos evaluados y el ambiente sobre el cual se está trabajando:

```
( DEFUN APLICAR ( FN LAE AMB )  
  ( IF ( ATOM FN ) ( COND  
    ( ( EQ FN `CAR ) ( CAAR LAE ) )  
    ( ( EQ FN `CDR ) ... )  
    ( ( EQ FN `CONS ) ( CONS ( NTH 0 LAE ) ( NTH 1 LAE ) ) )  
    ( ( EQ FN `+ ) ... )  
    ...  
    ... (aquí van las primitivas que vimos en clase, ej: -, APPLY, MAPCAR,...)  
    ...  
    ( T ( APLICAR ( EVAL FN AMB ) LAE AMB ) )  
  ) ( EVAL ( NTH 2 FN ) ( EXTENDERAMB ( NTH 1 FN ) LAE AMB ) ) )
```

Notas:

- La expresión: ( T ( APLICAR ( EVAL FN AMB ) LAE AMB ) ) evalúa una función definida por el usuario
- La expresión: ( EVAL ( NTH 2 FN ) ( EXTENDERAMB ( NTH 1 FN ) LAE AMB ) ) es una función LAMBDA (se evalúa en el ambiente)