

Gael Balderrama Dominguez
Tarea-07
Noviembre 2023

1.-Extiende el lenguaje agregando un nuevo operador *minu* que toma como argumento *n* y regresa $-n$. Por ejemplo, el valor de `minus(-(minus(5),9))` debe ser 14

Respuesta:

Especificacion Lexica :

Minus = Minus

Especificacion Sintactica :

Concreta : `Expression \Rightarrow minus (Expression)`

Abstracta : `(minus-exp exp1)`

Especificacion Semantica :

$$\frac{\mathcal{E}}{\mathcal{E}}$$

2.-Extiende el lenguaje agregando operadores para la suma, multiplicacion y conciente de enteros.

Respuesta:

3.- Agrega un predicado de igualdad numerica *equal?* y predicados de orden *greater?* y *less?* al conjunto de operaciones de lenguajes LET. Respuesta:

4.- Agrega operaciones de procesamiento de listas al lenguaje, incluyendo *cons*, *car*, *cdr*, *null?* y *emptylist*. Una lista debe poder contener cualquier valor expresado, incluyendo otra lista Respuesta:

5.- Agrega una operacion *list* al lenguaje. Esta operacion debe tomar cualquier cantidad de argumentos y regresar un valor expresado de la lista de sus valores. Respuesta:

7.-Incorpora al lenguaje expresiones *cond*. Usa la gramatica

$$Expression \Rightarrow condExpression \Rightarrow Expression * end$$

En esta expresión, las expresiones de los lados izquierdos de los \Rightarrow son evaluadas en orden hasta que una de ellas regresa un valor verdadero. Entonces el valor de toda la expresión es el valor de la expresión correspondiente al lado derecho

de esa \Rightarrow . Si ninguno de los lados izquierdos es verdadero, la expresión debe reportar un error.

Respuesta:

8.- Cambia los valores del lenguaje para que los enteros sean los únicos valores expresados. Modifica *if* para que el valor de 0 sea tratado como falso y todos los otros sean tratados como verdaderos. Modifica los predicados de manera consistente.

Respuesta:

9.- Como una alternativa al ejercicio anterior, agrega una nueva categoría sintáctica *Bool - exp* de expresiones booleanas al lenguaje. Cambia la producción para expresiones condicionales para que sea

$$Expression \Rightarrow \text{if } Bool - exp \text{ then } Expression \text{ else } Expression$$

Escribe Producciones apropiadas para *Bool - exp* y especifica su semántica con *Value - of - bool - exp* (Puedes abreviarlo como *B*). ¿En dónde terminan estando los predicados del ejercicio 3 con este cambio?

Respuesta:

10.- Modifica la implementación del intérprete agregando una nueva operación *print* que tome un argumento, lo imprime y regresa el entero 1. ¿Por qué esta operación no es expresable en nuestro método de especificación formal?

Respuesta:

11.- Extiende el lenguaje para que las expresiones *let* puedan vincular una cantidad arbitraria de variables, usando la producción,

$$Expression \Rightarrow \text{let } \{identifier = Expression^*\} \text{ in } Expression$$

Respuesta:

12.- Extiende el lenguaje con una expresión *Let** que funciona como en racket.

Respuesta:

13.- Agrega una expresión al lenguaje de acuerdo a la siguiente regla

$$Expression \Rightarrow \text{unpack } \{identifier\}^* = Expression \text{ in } Expression$$

tal que *unpack x y z = lst in ...* vincula *x*, *y* y *z* a los elementos de *lst* si *lst* es una lista con exactamente tres elementos, reportando un error en otro caso.

Respuesta: