



Software Design Document

Matemáticas Computacionales

Luis Bodart A01635000
17-5-2021

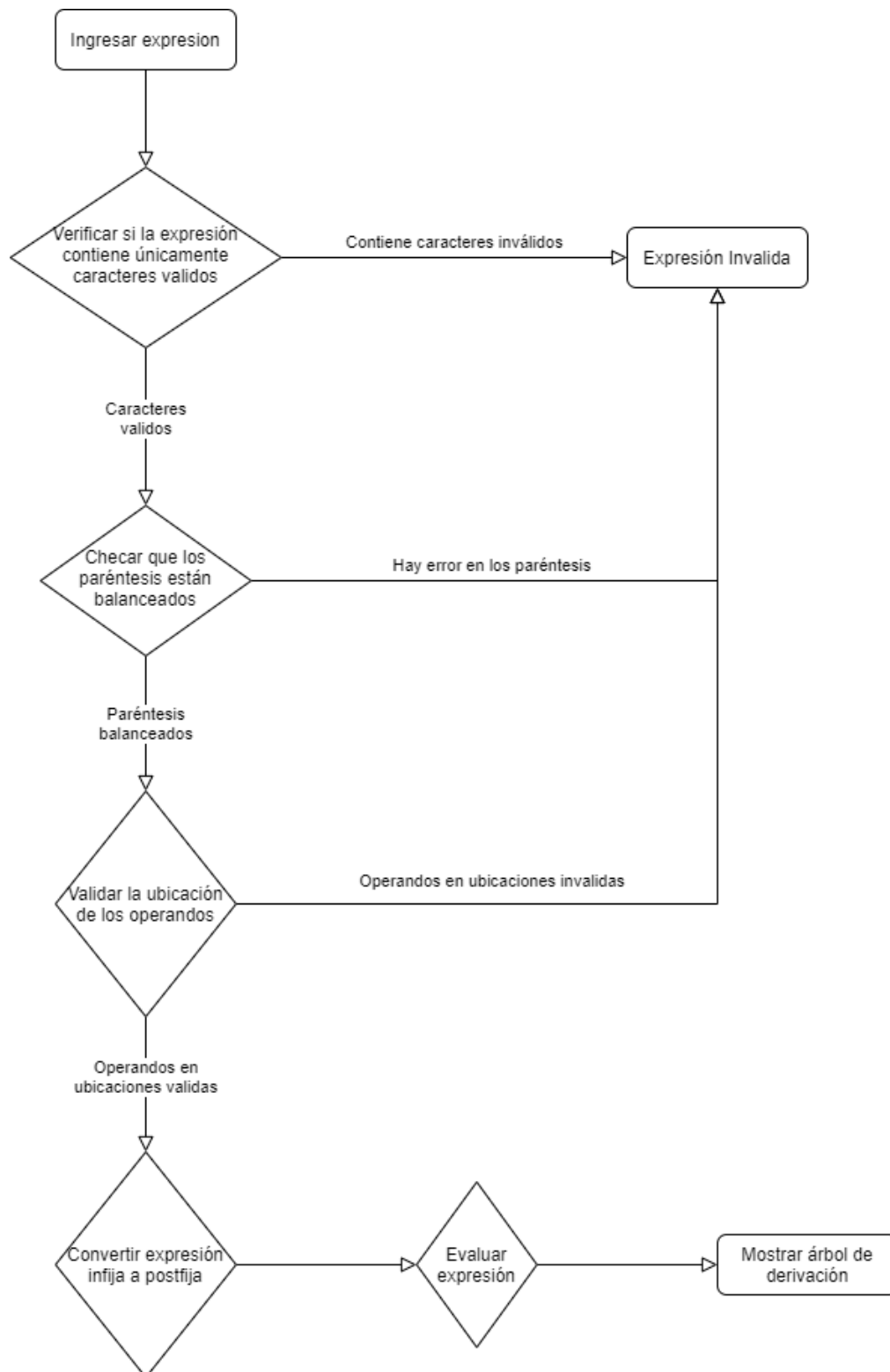
Índice

CONTENIDO

Diagramas	2
Vista General	2
Algoritmo	3
Validar la expresión	3
Convertir expresión a postfija	3
Prioridad de los operandos	3
Evaluar expresión postfija	3
Mostrar el árbol de derivación	4
Dibujar los nodos del árbol	4

DIAGRAMAS

VISTA GENERAL



ALGORITMO

VALIDAR LA EXPRESIÓN

validaExpr()

1. Checar que la expresión no este vacía o sea de puros espacios.
2. Purgar todos los espacios de la expresión
3. Buscar todos los caracteres válidos.
 - a. Si el tamaño de los caracteres validos es diferente al tamaño de la expresión, contiene caracteres inválidos.
4. Checar que los paréntesis de la expresión estén balanceados.
 - a. Si hay un paréntesis de cierre y el stack esta vacío, hay error en los paréntesis.
 - b. Si al final el stack no está vacío, faltan paréntesis de cierre.
5. Validar la ubicación de los operadores
 - a. Si al inicio de la expresión hay un '*', es una expresión invalida.
 - b. Si al final de la expresión hay un '+', '-', o '**', es una expresión invalida.
 - c. Si en la expresión hay un '+' o '-' y el carácter anterior es un '+' o '-', es una expresión invalida.
 - d. Si en la expresión hay un '*' y el carácter anterior es un '+', '-', o '**', es una expresión invalida.

CONVERTIR EXPRESIÓN A POSTFIJA

convertirPostfija(String infija)

1. Mantener los números que tienen más de un digito juntos.
2. Convierte la expresión a postfija de acuerdo con la jerarquía de los operandos.

PRIORIDAD DE LOS OPERANDOS

prioridad(Char token)

1. Checa la prioridad del token que recibe.
 - a. Prioridad: numero, '*', '+', '-', '(', ')', '(', '('.

EVALUAR EXPRESIÓN POSTFIJA

evaluarPostfija(Stack postfija)

1. Convierte el stack postfija en lista.
2. Se crea un nuevo stack para manipular el orden de evaluación.
3. Evalúa la expresión postfija
 - a. Si el token es un digito, se agrega al stack.
 - b. Si el token es un '+'.
 - a. Si el token es un '+', se agrega al stack.

- i. Si el tamaño del stack es menor a 1, saca un número del stack y guarda el resultado en el stack.
 - ii. Saca dos números del stack y los suma guardando el resultado en el stack.
 - c. Si el token es un '-'.
 - i. Si el tamaño del stack es menor a 1, saca un número del stack y guarda el resultado negativo en el stack.
 - ii. Saca dos números del stack y los resta guardando el resultado en el stack.
 - d. Si el token es un '*'.
 - i. Si el tamaño del stack es menor a 1.
 1. Si el token dos posiciones después es '+', saca un número del stack y guarda el resultado en el stack, se mantiene la posición para volver a checar.
 2. Si el token dos posiciones después es '-', saca un número del stack y guarda el resultado negativo en el stack, se mantiene la posición para volver a checar.
 - ii. Saca dos números del stack y los multiplica guardando el resultado en el stack.
4. El ultimo valor que quedo en el stack es el resultado final.

MOSTRAR EL ÁRBOL DE DERIVACIÓN

arbolDerivacion(List tokens, float resultado)

1. Muestra la expresión, la expresión postfija y el resultado.
2. Usa la clase Árbol para ordenar los tokens en Nodos, guardando la raíz del árbol completo.
 - a. Un nodo tiene valor, hijo izquierdo, hijo derecho, posición en x y posición en y.
3. Saca el nivel de profundidad máximo que tiene el árbol.
 - a. Si el nivel es de 1 dibuja directamente el valor de la raíz.
 - b. Si el nivel es de 2 dibuja el valor de la raíz.
 - i. Si tiene hijo izquierdo dibuja el valor del hijo izquierdo y una línea que conecta el nodo con el hijo izquierdo.
 - ii. Si tiene hijo derecho dibuja el valor del hijo derecho y una línea que conecta el nodo con el hijo derecho.
 - c. Si el nivel es mayor a 6 avisa que el árbol de derivación es muy grande para dibujarlo.

DIBUJAR LOS NODOS DEL ÁRBOL

dibujarArbol(Nodo nodo, int nivel)

1. Si el nodo es null regresa.
2. Si el nodo tiene hijo izquierdo e hijo derecho es padre.

- a. Si el nodo falta por dibujar, dibuja el valor del nodo.
- 3. Si el nodo tiene hijo izquierdo
 - a. Si el nodo falta por dibujar.
 - i. Si el nodo izquierdo no tiene hijos dibuja el valor del nodo izquierdo, una línea que conecta el nodo con el hijo izquierdo y marca el nodo izquierdo como dibujado.
 - ii. Si el nodo izquierdo tiene hijos, dibuja solo la línea que conecta el nodo con el hijo izquierdo.
- 4. Si el nodo tiene hijo derecho
 - a. Si el nodo falta por dibujar.
 - i. Si el nodo derecho no tiene hijos dibuja el valor del nodo derecho, una línea que conecta el nodo con el hijo derecho y marca el nodo derecho como dibujado.
 - ii. Si el nodo derecho tiene hijos, dibuja solo la línea que conecta el nodo con el hijo derecho.
- 5. Se llama otra vez la función usando el nodo izquierdo y el nivel $- 1$
- 6. Se llama otra vez la función usando el nodo derecho y el nivel $- 1$