

## Tarea 3: Cálculo de derivadas usando árboles binarios

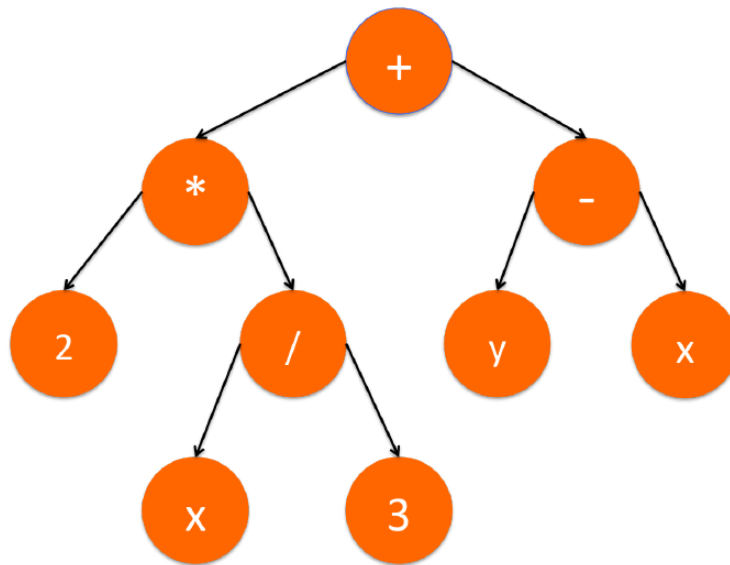
Profesores: Nelson Baloian, Jérémy Barbay, Patricio Poblete  
Fecha de entrega: 18-05-2018

### 1 Introducción

Como se vio en clases, un árbol binario se puede usar para guardar expresiones aritméticas. Además de la ventaja de que es fácil evaluar una expresión cuando está representada por un árbol, otra ventaja es que permite su manipulación simbólica, por ejemplo, cuando se quiere derivar la expresión almacenada.

Esta tarea se trata precisamente de construir un árbol de expresión aritmética a partir de una expresión textual (contenida en un String) escrita en notación polaca inversa, calcular su derivada respecto de una variable dada y publicar el resultado.

Ejemplo de construcción de árbol: si la expresión es  $2 \times 3 / * y x - +$ , el árbol de expresión que se debe construir será:



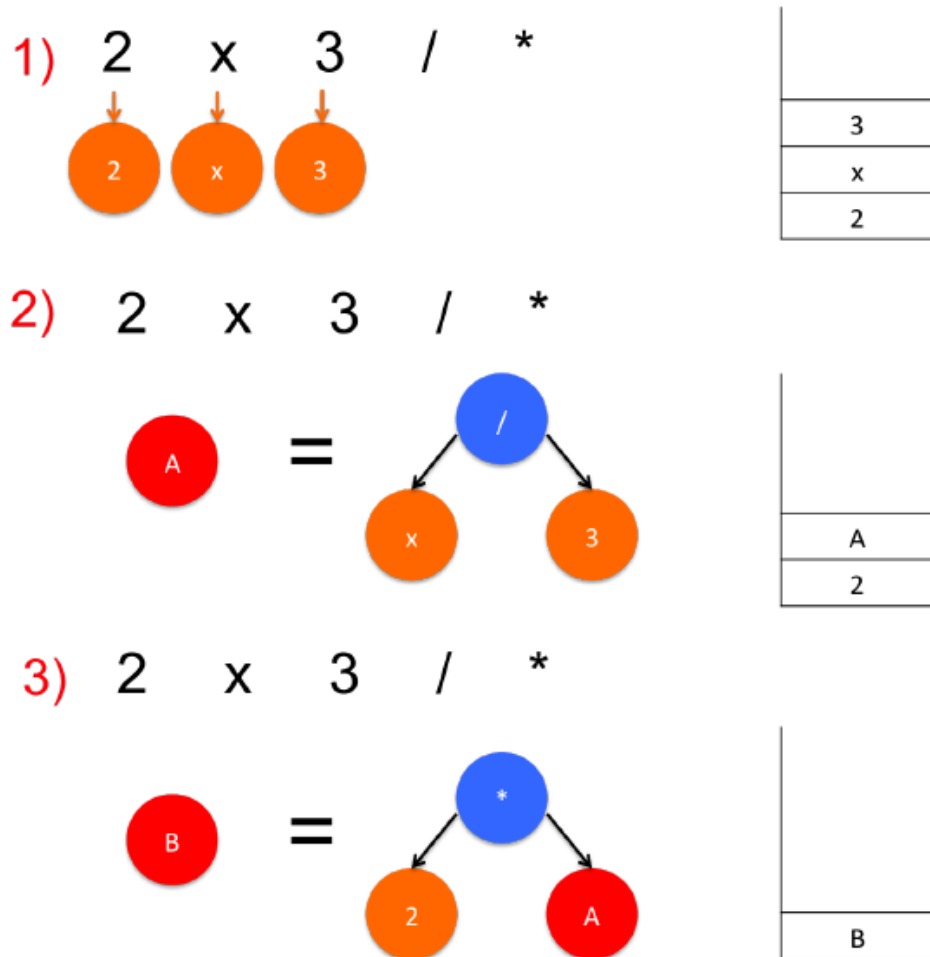
### 2 Explicación

Para obtener el árbol binario que represente la expresión, se debe construir una pila de nodos convenientemente definidos tal que al final del algoritmo la pila contenga un solo elemento que es precisamente el árbol a encontrar.

- Si el símbolo que se lee es un número o una variable, se crea un nodo con ese valor y se hace **push** a la pila.

- Si el símbolo que se lee es una operación, entonces se crea un nodo, con el valor de esa operación, tal que sus hijos son los elementos del **stack** que le siguen. El siguiente elemento será su nodo derecho y el otro el izquierdo.

Un ejemplo se visualiza en la siguiente figura, donde finalmente B es el árbol de la expresión:



Para construir un nuevo árbol que contenga la expresión de la derivada de la expresión original basta aplicar las reglas de la derivación, las cuales son recursivas:

- Si el nodo raíz contiene un número o una variable que no es respecto a la cual se deriva la expresión, el resultado es un nodo que contiene un cero.
- Si el nodo raíz contiene la variable respecto a la cual se deriva la expresión el resultado es un nodo con el número 1
- Si el nodo raíz contiene una operación entonces el resultado es un árbol que contiene la expresión correspondiente a la regla de la derivada para la operación: por ejemplo, para la suma es un árbol con un nodo raíz que contiene un operador de suma, el hijo izquierdo es un sub-árbol que contiene la derivada para la expresión de la izquierda y el hijo derecho es un sub-árbol que contiene la expresión de la derivada para la expresión contenida en el sub-árbol derecho.

### 3 Implementación

Se le pide que haga un programa que lea desde el teclado primero una expresión en polaca reversa, luego el nombre de la variable con respecto a la cual se debe derivar y responda con la expresión en in-fijo (notación aritmética normal) de dicha expresión. Esta última debe cumplir con las siguientes simplificaciones:

- Las expresiones tendrán las siguientes operaciones: suma, resta, multiplicación y división.
- Los símbolos serán +, -, \* y / respectivamente.
- Las multiplicaciones de un término por 0 se deben reemplazar por 0.
- Las multiplicaciones de un término por 1 se deben reemplazar por el mismo término.
- Las sumas o restas de un término con 0 se deben reemplazar por el mismo término.
- Las divisiones de un término por 1 se deben reemplazar por el término.
- La expresión final debe omitir los paréntesis que no son necesarios. Por ejemplo:  
 $((a * b) * c) + d) / e = (a * b * c + d) / e$

Un ejemplo de salida simplificada será:  $(2 * x) + 1 + 0 = 2 * x + 1$ . Para esto, su programa debe seguir la siguiente dinámica:

1. Construir el árbol de la expresión utilizando una pila, como se explicó en la sección anterior.
2. Construir el árbol de la expresión derivada.
3. Simplificar el árbol de la expresión resultante.
4. Recorrer este último convenientemente para imprimir la expresión solicitada.

Para simplificar el problema puede suponer lo siguiente:

- Los nombres de variables siempre tienen un solo carácter.
- Los números de la expresión serán números del 0 al 9.
- Los símbolos vienen siempre separados por un solo espacio.
- Las fórmulas de derivación son:

$$1. (f \pm g)' = f' \pm g'$$

$$2. (fg)' = f'g + fg'$$

$$3. \left(\frac{f}{g}\right)' = \left(\frac{f'g - fg'}{g * g}\right)$$

### 4 Ejemplos de entrada y salida

Un ejemplo de entrada y salida es el siguiente:

Entrada: 2 x 3 / \* y x - +

Lo cual corresponde a la expresión:  $2 * (x/3) + (y + x)$

El árbol de esta expresión es el ilustrado en la sección de Introducción.

Las salidas son:

Salida derivando x:  $(2 * (1/3)) + (0 + 1) = 2 * 1/3 - 1$

Salida derivando y:  $(0 + 0) + (1 - 0) = 1$

## 5 Condiciones de Entrega

- Esta tarea debe ser resuelta en Java.
- Es obligatoria la entrega de un informe junto con su tarea (Ver siguiente sección).
- Esta tarea es de carácter individual, cualquier caso de copia se evaluará con la nota mínima.
- No olvide subir a U-cursos todos los archivos necesarios para que su tarea funcione correctamente.
- Debe subir los archivos de código fuente (\*.java). Los archivos compilados (\*.class) no serán evaluados.

Cualquier duda respecto a la tarea puede ser consultada usando el foro del curso.  
No se aceptarán atrasos.

## 6 Informe

El informe debe describir el trabajo realizado, la solución implementada, los resultados obtenidos y las conclusiones o interpretaciones de estos. Principalmente debe ser breve, describiendo cada uno de los puntos que a continuación se indican:

- **Portada:** Indicando número de la tarea, fecha, autor, email, código del curso, etc.
- **Introducción:** Descripción breve del problema y su solución.
- **Análisis del problema:** Exponga en detalle el problema, los supuestos que pretende ocupar, casos de borde y brevemente la metodología usada para resolverlo.
- **Solución del problema:** Indique claramente los pasos que siguió para llegar a la solución del problema. Muestre mediante figuras y ejemplos qué es lo que realiza su código. Evite copiar todo el código fuente en el informe, sin embargo, puede mostrar las partes relevantes de este.
- **Modo de uso:** Explicando brevemente cualquier dato necesario para la compilación y ejecución de su programa.
- **Resultados:** Muestre ejemplos de entradas y salidas de su programa. Además muestre el resultado de los experimentos, por ejemplo, la expresión de entrada y la de salida.