Gerald Francisco Mora Mora
2014064955
CE-1103

Arboles Rojo y Negro

Este es un árbol Binario estricto, en el cual todo nodo hoja es nulo y se presentan en color de estado negro. Además cada Nodo tiene un color de estado, ya sea rojo o puede ser negro. En este tipo de árbol la raíz posee color de estado negro. Para implementarse debe cumplir las siguientes condiciones:
1º- Un nodo rojo tiene dos hijos negros.
2º- Todo camino de la raíz a cualquier hoja pasa por el mismo numero de nodos negros.
Para n que sea un nodo se le puede calcular su altura de la siguiente manera:

$$\mathbf{H}(n) = \begin{cases} \max\big(\mathbf{H}(n.izdo), \mathbf{H}(n.dcho)\big) + 1 & \text{si } n \text{ es } \textbf{negro} \\ \max\big(\mathbf{H}(n.izdo), \mathbf{H}(n.dcho)\big) & \text{si } n \text{ es } \textbf{rojo} \end{cases}$$

Figura 1. Formula para el calculo de la altura de un nodo n en un árbol rojo y negro.

Propiedades:
_ Si se cambia un nodo de rojo a negro no afecta la condición uno, pero si lo hace con la condición 2.
_ Cambiar un nodo de negro a rojo si puede afectar a la condición 1 y también así puede afectar a la segunda condición.
_ Si se da que como resultado de una operación la raíz pasa a ser rojo, esta se puede cambiar a negro de forma directa sin afectar a las dos condiciones.
_ Se puede borrar un nodo rojo sin afectar las condiciones, pero no ocurre así al borrar un nodo negro.

Inserción:
Inicialmente es igual a la de un árbol binario de búsqueda. Para no violar la segunda condición el nuevo nodo tendrá color rojo. Si se cumple que el padre es de color negro, no se va a realizar ningún ajuste. Para el caso contrario se entra en un bucle donde **x** representa el nodo que se está comprobando. **x** es un nodo rojo, y los casos se refieren a situaciones en que su padre existe y es también rojo: Cuando el padre no exista, simplemente se cambia el color de **x** (que va a ser la raíz) a negro y se termina el bucle, y si el padre existe y es negro se termina directamente el bucle.

Casos no triviales:

Caso 1: Tío rojo, nodo x izquierdo o derecho:
Solución: Se cambia de color a los nodos y, z y t. Puede darse el problema de que el padre de z sea rojo y se viole de nuevo la primera condición. Iteración siguiente: El nodo que se comprueba ahora es z.
Caso 2: Tío negro, nodo x es hijo derecho:
Solución: Se efectúa una rotación x-y. Esto no resuelve el problema, pero hace que ahora el nodo que hay que comprobar sea un hijo izquierdo y por lo tanto se aplique el caso siguiente. Iteración siguiente: El nodo que se comprueba ahora es **y.** Se caerá en el Caso 3.

Caso 3: Tío negro, nodo x es hijo izquierdo:

Solución: Se efectúa una rotación z-y, se cambia de color a **z** e **y.** Esto resuelve el problema. Iteración siguiente: El árbol cumple las condiciones. Se termina el bucle.

Borrado en arboles Rojo y Negros:

Este se lleva a cabo de la misma manera que en los arboles binarios de búsqueda. Se busca el nodo a borrar(si este existe, este debe tener siempre dos hijos). Si es un nodo con dos hijos no nulos, se busca el nodo mayor de su subárbol izquierdo, se van a intercambiar sus datos con el nodo a borrar y se pasa a borrar el nodo **x**. Y como este es el nodo mas a la derecha, su hijo derecho será nulo.

Casos Triviales:

_ Nodo borrado es rojo: El árbol sigue siendo rojo-negro, por lo que no hay que realizar ningún ajuste. Se considera que el nodo borrado es negro.

_ El hijo, x, es rojo: En este caso se incumple la condición de que los hijos de z tengan la misma altura negra. Cambiando el color de x a negro se restablece la condición.

Casos imposibles:

Caso 1. Hermano rojo, padre negro:

Solución: Se hace una rotación padre-hermano y se cambian sus colores. El nodo **x** pasa a tener como nuevo hermano al nodo **a**, sigue teniendo una altura menor en uno que la de su hermano, pero ahora su padre es rojo y por lo tanto caerá en alguno de los siguientes casos (dependiendo del color de **a**): Caso 3, 4 o 5. Iteración siguiente: Se vuelve a comprobar con los mismo nodos **x** y **z**, se garantiza que no se cae en éste caso.

Caso 2. (Hermano negro no nulo, sobrinos negros, padre negro):

Solución: Se cambia el color del hermano (**y**) a rojo. Con ello los nodos **x** e **y** pasan a tener la misma altura negra. El problema es que la altura de **z** ha disminuido. Iteración siguiente: Se vuelve a comprobar, ahora el nodo llamado **x** es el nodo **z** y el nodo llamado **z** es el padre de **z**.

Caso 3: (Hermano negro no nulo, sobrinos negros, padre rojo):

Solución: Se cambia el color del hermano (**y**) a rojo y el del padre (**z**) a negro. Con ello los nodos **x** e **y** pasan a tener la misma altura negra. La altura de **z** sigue siendo la misma. Iteración siguiente: El arbol ya cumple todas las condiciones. Se termina el bucle.

Caso 4: (Hermano negro no nulo, sobrinos rojo/negro, padre cualquier color):

Solución: Se realiza una rotación hermano-sobrino izquierdo y se cambian sus colores. Los hijos del sobrino izquierdo existen (aunque pueden ser nulos) y son negros, ya que el sobrino izquierdo es rojo. El nodo **x** pasa a tener como hermano al nodo **a** y sigue teniendo una altura negra menor en uno que la de su hermano. Iteración siguiente: Nuevamente se comprueba nodos x y z y se acera al caso 5.

Caso 5: (Hermano negro no nulo, sobrinos cualquiera/rojo, padre cualquier color):

Solución: Se realiza una rotación padre-hermano y se cambia el color de la siguiente forma: El padre (z) pasa a ser negro, el hermano (y) toma el mismo color que el que tenía originalmente z, el sobrino derecho pasa de rojo a negro (este sobrino debía existir ya que era rojo). Iteración siguiente: El árbol ya cumple todas las condiciones. Se termina el ajuste.