Árboles

Dr. Said P. Martagón

Universidad Politécnica de Victoria

4 de abril de 2016

Outline

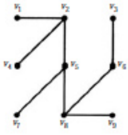
Árboles

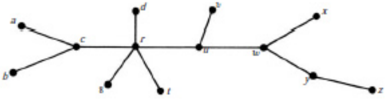
Árboles con raíz

Árboles binarios

Árboles

Un grafo T se denomina árbol si T es conexo y T no tiene ciclos. En la figura se muestran ejemplos de árboles. Un *bosque* G es un grafo sin ciclos; por tanto, los componentes conexos de un bosque G son árboles. Un grafo sin ciclos es *libre de ciclos*. El árbol que consta de un solo vértice sin aristas se denomina *árbol degenerado*.





Considere un árbol T. Resulta evidente que sólo hay un camino simple entre dos vértices de T; en caso contrario, los dos caminos formarían un ciclo. También:

- Suponga que en T no hay ninguna arista $\{u, v\}$ y que a T se agrega la arista $e = \{u, v\}$. Entonces el camino simple de u a v en T y e forma un ciclo; por lo tanto, T ya no es un árbol.
- Por otra parte, suponga que en T hay una arista $e = \{u, v\}$, que de T se elimina e. Entonces T ya no es conexo (puesto que no puede haber ningún camino de u a v); así, T ya no es un árbol.

Teorema

Sea G un grafo con $n > 1$ vértices. Entonces las siguientes afirmaciones son equivalentes:

I) G es un árbol.

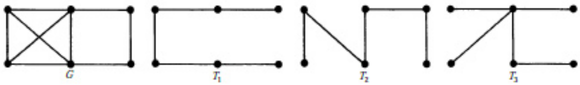
II) G es libre de ciclos y tiene $n - 1$ aristas.

III) G es conexo y tiene $n - 1$ aristas.

Este teorema indica que un árbol finito con n vértices debe tener $n - 1$ aristas.

Outline

Un subgrafo T de un grafo conexo G se denota *árbol de expansión* de G si T es un árbol y T incluye a todos los vértices de G. En la figura se muestra un grafo conexo G y árboles de expansión T_1, T_2 y T_3 de G.



Outline

Suponga que G es un grafo ponderado conexo. Es decir, a cada arista de G se asigna un número no negativo denominado *peso* de la arista. Entonces a cualquier árbol de expansión T de G se asigna un peso total que resulta de sumar los pesos de las arístas en T. Un *árbol de expansión mínima* de G es un árbol de expansión cuyo peso total es el más pequeño posible.

Algoritmo (1)

La entrada es un grafo ponderado conexo G con n vértices.

1. Las aristas de G se disponen en orden decreciente de peso.
2. Se procede secuencialmente para eliminar cada arista que no haga inconexo al grafo, hasta que quede n-1 aristas.
3. Salir.

Algoritmo de Kruskal (2)

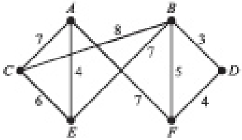
La entrada es un grafo ponderado conexo G con n vértices.

1. Las aristas de G se disponen en orden creciente de peso.
2. Se empiesa sólo con los vértices de G y en forma secuencial se agrega cada arista que no origine un ciclo hasta que se hayan agregado $n-1$ aristas.
3. Salir

El peso de un árbol de expansión mínima es único, aunque el árbol de expansión mínima en si no lo es. Cuando dos o más aristas tienen el mismo peso pueden ocurrir distintos árboles de expansión mínima. En este caso, la disposición de las aristas en el paso 1 de los algoritmos anteriores no es única, y así resultan árboles de expansión mínima distintos.

Ejemplo

Encontrar un árbol de expansión mínimo del grafo ponderado Q. Observe que Q tiene seis vértices, de modo que un árbol de expansión mínima tiene cinco aristas.



Usando el algoritmo (1)

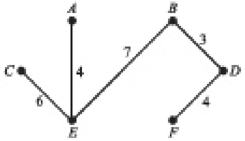
Primero se ordenan las aristas en orden decreciente de peso y luego en forma consecutiva se eliminan las aristas sin hacer inconexo a Q hasta que queden cinco aristas. Así se obtienen los datos siguientes:

Aristas	BC	AF	AC	BE	CE	BF	AE	DF	BD
Peso	8	7	7	7	6	5	4	4	3
Eliminar	Si	Si	Si	No	No	Si			

Así, el árbol de expansión mínima de Q que se obtiene contiene las aristas

$$BE \quad CE \quad AE \quad DF \quad BD$$

El peso del árbol de expansión es 24 y se muestra en la figura.



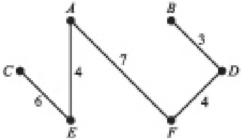
Aplicando el algoritmo (2)

Primero se ordenan las aristas en orden creciente de peso y enseguida se agregan las aristas sin formar ningún ciclo hasta que se incluyen cinco arista. Así se obtienen los datos siguientes:

Aristas	BD	AE	DF	BF	CE	AC	AF	BE	BC
Peso	3	4	4	5	6	7	7	7	8
¿Agregar?	Si	Si	Si	No	Si	No	Si		

Así, el árbol de expansión mínima de Q contiene las aristas

BD AE DF CE AF



Así, el árbol de expansión se muestra en la figura. Observe que este árbol de expansión no es el mismo que se obtuvo al usar el algoritmo 1, y que, como era de esperar, su peso también es 24.

Observación

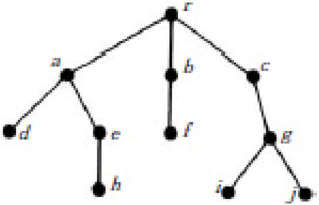
Los algoritmos anteriores se ejecutan fácilmente cuando el grafo G es relativametne pequeño, como en el caso del ejemplo. Suponga que G tiene docenas de bértices y centenas de aristas que, por ejemplo, se proporcionan mediante una lista de pares de vértices. Entonces decidir si G es conexo no es evidente; puede ser necesario algún tipo de algoritmo de búsqueda en profundidad en grafos (DFS: Deep-first search) o de búsqueda en anchura (BFS: Breadth-first search) en grafos.

Outline

Recuerde que un grafo de un árbol es un grafo conexo libre de ciclos; es decir, un grafo conexo sin ningún ciclo. Un árbol T con raíz es un grafo de un árbol con un vértice designado r al que se le denomina raíz del árbol. Puesto que existe un camino simple único de la raíz r a cualquier otro vértice v en T, esto determina una dirección hacia las aristas de T. Por tanto, T puede considerarse como un grafo dirigido. Observe que cualquier árbol puede hacerse un árbol con raíz con la simple elección de uno de los vértices como la raíz.

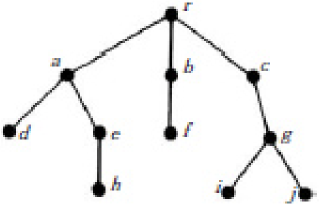
Considere un árbol T con raíz cuya raíz es R. La longitud del camino de la raíz r a cualquier vértice v se denomina nivel (o profundidad) de v, y el máximo nivel de vértice se denomina profundidad del árbol. Los vértices con grado 1, que no sean la raíz r, se denominan *hojas* de T, y un camino dirigido de un vértice a una hoja se denomina *rama*.

Por lo general, un árbol T con raíz se ilustra con la raíz en la parte superior del árbol. En la figura se muestra un árbol T con raíz r y 10 vértices más. El árbol tiene cinco hojas d, f, h, i y j. Observe que $nivel(a) = 1$, $nivel(f) = 2$, $nivel(j) = 3$. Además, la profundidad del árbol es 3.



El hecho de que un árbol T con raíz proporcione una dirección a las aristas significa que es posible asignar una relación de precedencia entre los vértices. De manera más precisa, se dice que un vértice u precede a un vértice v o que v sigue a u si hay un camino (dirigido) de v a u. En particular, se dice que v *sigue inmediatamente de u si* (u, v) es una arista; es decir, si v sigue a u y v es adyacente a u.

Observe que cualquier vértice v, que no sea la raíz, sigue inmediatamente a un vértice único, aunque v puede ser seguido inmediatamente por más de un vértice. Por ejemplo, en la figura siguiente, el vértice j sigue a c aunque inmediatamente sigue a g. Ambos vértices i y j siguen inmediatamente a g.

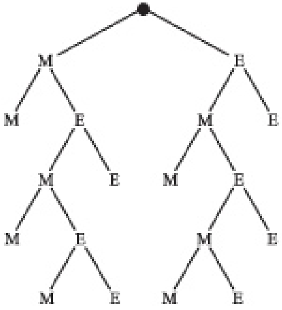


Un árbol T con raíz tambien es un dispositivo útil para enumerar todas las posibilidades lógicas de una secuencia de eventos donde cada evento puede ocurrir en una forma finita de formas. En este hecho en el siguiente ejemplo.

Ejemplo

Suponga que Marcos y Eric juegan en un torneo de tenis de modo que la primera persona en ganar dos jjuegos seguidos o quien gane un total de tres juegos gana el torneo. Encuentre el número de formas en que puede desarrollarse el torneo.

El árbol con raíz en la figura muestra las diferentes formas en que puede desarrollarse el torneo. Hay 10 hojas que corresponden a las 10 formas en que puede ocurrir el torneo.



Hay 10 hojas que corresponden a las 10 formas en que pueden ocurrir el torneo

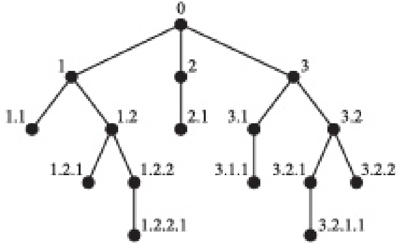
MM	MEMM	MEMEM	MEMEE	MEE	EMM	EMEMM	EMEME	EMEE	EE

Específicamente, el camino de la raíz a la hoja describe quién ganó cuáles juegos en el torneo.

Outline

Considere un árbol T con raíz en el que las aristas que salen de cada vértice están ordenadas. Entonces se tiene el concepto de *árbol con raíz ordenado.* Los vértices de un árbol así pueden etiquetarse (o direccionarse) en forma sistemática como: primero se asigna 0 a la raíz r. Luego se asigna $1, 2, 3, \ldots,$ a los vértices que siguen de inmediato a r según la forma en que se ordenaron las aristas.

Enseguida se etiquetan los vértices restantes: si a es la etiqueta de un vértice v entonces a,1, a,2, . . . se asignan a los vértices que siguen de inmediato a v según la forma en que se ordenaron las aristas. Este sistema de direcciones se ilustra en la siguiente figura.



Se observa en la figura anterior que las aristas se representan de izquierda a derecha según su orden. Observe que el número de puntos decimales en cualquier etiqueta es uno menos que el nivel del vértice. Este sistema de identificación se denomina *sistema universal de direcciones* para un árbol con raíz ordenado.

El sistema universal de direcciones constituye una forma importante para describir (o almacenar) linealmente un árbol con raíz ordenado. De manera más concisa, dadas las direcciones a y b, se hace $a < b$ si $b = a.c$ (es decir, a es un segmento inicial de b), o si hay enteros positivos m y n con $m < n$ tales que

$$a = r.m.s \quad \text{y} \quad b = r.n.t$$

Este orden se denomina *orden lexicográfico* puesto que es semejante a la forma en que las palabras están dispuestas en un diccionario. Por ejemplo, las direcciones en la figura anterior están ordenadas linealmente según se representa en la figura siguiente. Este orden lexicográfico es idéntico al orden que se obtiene al moverse hacia abajo a partir de la rama más a la izquierda del árbol, en seguida hacia la siguiente rama a la derecha, luego la segunda rama a la derecha y así sucesivamente.

0	2.1
1	3
1.1	3.1
1.2	3.1.1
1.2.1	3.2
1.2.2	3.2.1
1.2.2.1	3.2.1.1
2	3.2.2

Outline

Introducción

El árbol binario es una estructura fundamental en matemáticas y computación y también se le aplican algunos de los términos de los árboles con raíz como arísta, camino, rama, hoja, profundidad y número de nivel. No obstante, en los árboles binarios se usará el término nodo, en lugar de vértice. Debe tener en cuenta que un árbol binario no es un caso especial de un árbol con raíz; son entes matemáticos diferentes.

Outline

Un *árbol binario* T es un conjunto finito de elementos que se denominan *nodos*, tales que:

- T es vacío (*árbol nulo o árbol vacio*), o
- T contiene un nodo distintivo R, denotado *raíz* de T, y los nodos restantes de T forman un par ordenado de árboles binarios ajenos T_1 y T_2.

Si T contiene una raíz R, entonces los árboles T_1 y T_2 se denominan, respectivamente , subárbol izquierdo y subárbol derecho de R. Si T_1 no es vacío, entonces su raíz se denomina *sucesor izquierdo* de R; en forma semejante, si T_2 no es vacío, entonces su raíz se denomina *sucesor derecho* de R.

La definición anterior de un árbol binario T es recursiva, ya que T se define en términos de los subárboles binarios T_1 y T_2. Esto significa, en particular, que cualquier nodo N de T contiene un subárbol izquierdo y un subárbol derecho, y que cada subárbol o ambos pueden ser vacios. Así, cualquier nodo N en T tiene $0, 1 o 2$ sucesores. Un nodo sin sucesores se denomina nodo *terminal*. Por lo tanto, los dos subárboles de un nodo terminal son vacíos.

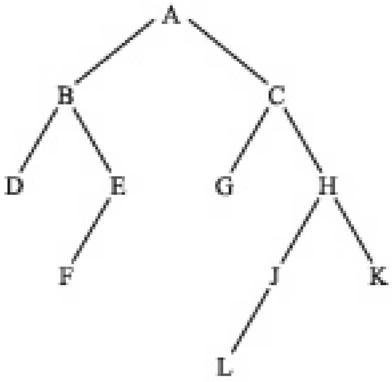
Representación de un árbol binario

Un árbol binario T suele representarse por medio de un diagrama en plano, denominado *ilustración* de T. En específico, el diagrama de la figura representa un árbol que:

I) T consta de 11 nodos, que se representan con las letas A a L, excepto la I.

II) La raíz de T es el nodo A en la parte superior del diagrama.

III) Una línea inclinada hacia la izquierda en un nodo T indica un sucesor izquierdo de N; y que una línea inclinada hacia la derecha en T indica un sucesor derecho de N.


A
B
C
D
E
G
H
F
J
K
L

Por consiguiente:

a) B es un sucesor izquierdo y C es un sucesor derecho de la raíz A.

b) El subárbol izquierdo de la raíz A consta de los nodos B, D, EyF, y el subárbol derecho consta de los nodos C, G, H, J, KyL.

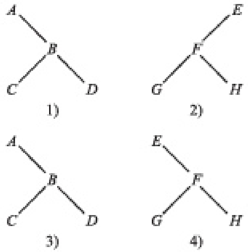
c) Los nodos A, B, CyH tienen dos sucesores; los nodos E y J tienen sólo un sucesor, y los nodos D, F, G, LyK no tienen sucesores; es decir, son nodos terminales.

Árboles binarios semejantes

Se dice que los árboles binarios TyT' son *semejantes* si tienen la misma estructura o, en otras palabras, si tienen la misma forma. Se dice que son *copias* si son semejantes y si tienen el mismo contenido en nodos correspondientes.

Ejemplo

Considere los cuatro árboles binarios en la figura. Los tres árboles 1), 3) y 4) son semejantes. En particular, los árboles 1) y 3) son copias, puesto que tienen los mismos datos en los nodos correspondientes. El árbol 2) no es semejante ni copia del árbol 4) porque, en un árbol binario, se distingue entre un sucesor izquierdo y un sucesor derecho incluso cuando sólo hay un sucesor.



Terminología

Para describir relaciones entre los nodos de un árbol T a menudo se usa la terminología que describe relaciones familiares: suponga que N es un nodo en T con su sucesor izquierdo S_1 y sucesor derecho S_2. Entonces N se denomina *padre* de S_1 y S_2. En forma semejante, S_1 se denomina *hijo izquierdo* de N y S_2 se denomina *hijo derecho* de N. Además, se dice que S_1 y S_2 son *hermanos*. Todo nodo N en un árbol binario T, excepto la raíz, tiene un padre único, denominado *predecesor* de N.

Terminología

Los elementos descendiente y ancestro tienen su significado de costumbre. Es decir, un nodo L se denomina *descendiente* de un nodo N (y N se denomia *ancestro de* L) si existe una sucesión de hijos de N a L; y se especifica si L es *descendiente izquierdo o derecho* de N según si L pertenece al subárbol izquierdo o derecho de N.

Terminología

La terminología de la teoría de grafos y de la horticultura también se usa con un árbol binario T. Para mayor claridad, la línea que se traza desde un nodo N de T hasta un sucesor de denomina *arista*, y una secuencia de aristas consecutivas se denomina *camino*. Un nodo terminal se denomina *hoja*, y un camino que termina en una hoja se denomina *rama.*

Terminología

A cada nodo en un árbol binario T se le asigna un *número de nivel* en el orden siguiente: a la raíz R del árbol T se le asigna el número de nivel 0, a los demás nodos se les asigna un número de niveles que es 1 más que el número de nivel de su padre. Además, se dice que los nodos con el mismo número de nivel pertenecen a la misma *generación*.

Terminología

La *profundidad (o altura)* de un árbol T es el número máximo de nodos en una rama de T. Resulta que ésta es una unidad mayor que el número de nivel de T. El árbol T en la figura tiene profundidad 5.